op5 Monitor administrator manual

op5 Monitor administrator manual©
Version 5.5 Rev 1.1.0

op5 Monitor administrator manual
Version 5.5, Rev 1.1
Author: Professional Services

op5 web site
www.op5.com

op5 support
www.op5.com/support

op5, and the op5 logo are trademarks, service marks,
registered service marks or registered trademarks of op5 AB

Contents

Introduction

About introduction

This chapter covers the following topics:

Using this manual

This manual includes information about the more advanced parts op5 Monitor and its components.

The manual is written with the goal to give the reader help about how to manage the different parts of op5 Monitor.

This manual is targeted for a technical audience. The manual covers how to use and configure op5 Monitor through its web interface. For configuration using direct console access or SSH, see the op5 System manual.

About op5 Monitor

op5 Monitor is a highly flexible monitoring system for monitoring of IT infrastructure. op5 Monitor is based on the widely known open source monitoring system Nagios.

op5 Monitor is used and configured in a web interface using any standard browser. The most common browsers Internet Explorer, Firefox and Opera have been tested.

The interface is protected by using both authentication (username and password) and by SSL which enables a secure manner for accessing the web interface using encryption.

Agents

About agents

This chapter covers the following topics:

Introduction

Most of the monitoring in op5 Monitor is used with the help of agents. The plugins are contacting the agents and let them do the job.

There are mainly four agents available for download at the op5 support site.

Agent	Environment
op5 NSClient++	Microsoft Windows
NRPE	Unix/Linux
MRTGEXT	Novell
Windows SyslogAgent	Microsoft Windows

op5 NSClient++

This is the agent used for monitoring Windows type operating systems.

This agent has the ability to function as a drop in replacement for NSClient providing the same features as NSClient combined with the ability to execute scripts on the monitored Windows server.

Installing and configuring op5 NSClient++

The installation and configuration of op5 NSClient++ is covered in detail in the op5 video tutorial **How to monitor windows servers**:

http://www.op5.com/support/documentation/video-tutorials

Plugins used with op5 NSClient++

There are mainly two plugins that is used to communicate with op5 NSClient++:

- check_nt
- check_nrpe

check_nt

This plugin is used for all basic tests like

- cpu
- memory
- disks

But it can also be used to check

- Windows services
- performances counters

check_nrpe

check_nrpe can also be used in the communication with op5 NSClient++. This one is normally used when you are performing checks on the Windows server with custom scripts.

Configuration files

NSClient++ operation is configured in a couple of plain text files called:

- NSC.ini
- op5.ini

- custom.ini

They are located in the install directory.

Table 1 Description of the configurations files

File	Description
NSC.ini	This is the standard configuration file. This contains the default settings for NSClient++  This file might be overwritten during an update of NSClient++
op5.ini	This is a op5 specific configuration file. Here are the changes made by op5 entered.  This file might be overwritten during an update of NSClient++
custom.ini	This is where you shall place your own files.  It will never be overwritten during any update of NSClient++.

The default configuration provided is fully functional but there are some options that likely need to be changed.

Changing the configuration

To change the configuration

To change the configuration open the `custom.ini` file using your favorite text-editor (e.g. WordPad). This file is empty and but take a look at NSC.ini to view all settings.
Read the `NSC.ini` file carefully to get a complete understanding of all configuration options. Lines starting with ; (semicolon) are comments.

Before the changes will take effect NSClient++ must be restarted.

Options most likely in need for configuration are described bellow, section by section.

```
[Settings]
allowed_hosts=
```

This option lists all servers that are allowed to talk to the agent. Enter the IP-address of the op5 Monitor/Statistics server. If this option is left blank anybody will be able to communicate with the agent.

```
[log]
debug=0
```

Set debug to `1` to enable debugging. This is normally not needed but can be very useful when debugging.

```
[NSClient]
port=1248
```

This is the port used for NSClient style requests, i.e. using the check_nt plugin. If any other application is already using the default port it might be necessary to change this option.

Note: If a non default port is used you also need to make changes on the op5 Monitor server.

```
[NRPE]
port=5666
```

This is the port used for nrpe style requests. In order for a minimum of configuration on the op5 Monitor server it's recommended that this option is left with the default value. If this is changed new nrpe check commands using the configured port need to be created on the op5 Monitor server.

```
allow_arguments=0
```

Set this to 1 to enable the possibility to include arguments in nrpe requests. This could be considered a security risk so only enable this if needed. Also, make sure to set the `allowed_hosts` option described above if arguments are allowed.

```
[NRPE Handlers]
```

The nrpe handlers provide a way to execute any custom plugin/check command on the monitored Windows server. In this section you configure all the commands that should be available.

Example 1 *Adding a custom script/plugin to NSClient++*

```
command[my_custom]=c:\mycustomdir\my_prog.exe
```

Or the simplified syntax:

```
my_custom=c:\mycustomdir\my_prog.exe
```

NRPE

NRPE is a Unix client for executing plugins on remote hosts.

It is distributed as

- rpm-packages
- deb-packages
- portable source-code.

NRPE is used in combination with a set of local plugins. By default in op5 Monitor the plugins are placed in:

```
/opt/plugins
```

There are only a few plugins shipped with the op5 NRPE packages but you may use the ones located on the op5 Monitor server.

Installing NRPE

To install NRPE

1. Download the package for your environment from the download section at the support site at www.op5.com
2. Put the package to the host you like to install it on.
3. Install the package the same way as you do normally with packages on that host.

Configuring NRPE

Before we can start use the NRPE agent for monitoring with op5 Monitor we need to configure the agent.

The NRPE agent is located in:

- `/etc/nrpe.conf`

Table 2 NRPE main configuration file settings

Setting	Description
`server_port`	The port NRPE should listen on. **Default**: `5666`
`allowed_hosts`	Add the IP of you OP5 Monitor server on this line multiple addresses can be separated with , ie: allowed_hosts=1.2.3.4,1.2.3.5  Make sure you do not add any space between the comma (,). **Default**: empty
`nrpe_user`	The user the NRPE daemon is executed as. **Default**: `nobody`
`nrpe_group`	The group the NRPE daemon is executed as. **Default**: `nobody`
`debug`	Set to `1` if you need to debug the NRPE. **Default**: `0`
`command_timeout`	The default time out, in seconds, a check shall have. **Default**: `60`
`dont_blame_nrpe`	Set to `1` to be able to send arguments to NRPE. **Default**: `0`

Adding commands to NRPE

NRPE comes with a few predefined commands. Those commands are located in:

`/etc/nrpe.d/op5_commands.cfg`

You may add your own commands and you should do that in your own file in:

`/etc/nrpe.d/`

You must set `.cfg` as extension to your configuration file or else it will not be loaded into NRPE when the daemon is restarted.

NRPE command definition

The NRPE command definitions is divided into two parts.

Table 3 NRPE command parts

Part	Description
`command[name]`	The string between the square brackets will be the name of this command. The name is used when you executes the command with check_plugin.  Do not use space in the command name.
`/opt/plugins/...`	This is the command line used to execute the plugin you are going to use in your command.

To add a command to NRPE

Here we will add a command that is looking for a process named smsd using the plugin check_procs, which is installed by default.

1. Login to the host you have installed NRPE on as root user over ssh.
2. Create a new configuration file and open it up with your favorite editor.
3. Add a command line looking like this:
   ```
   command[proc_smsd]=/opt/plugins/check_procs -w 1: -c 2:2 -C smsd
   ```
4. Save the file and restart NRPE:
   ```
   service nrpe restart
   ```

Plugins used with NRPE

The only plugin used with NRPE is

- check_nrpe

To use the plugin with the command defined in *Adding commands to NRPE* on page 11 you shall use the following command line in your service definition:

```
/opt/plugins/check_nrpe -H $HOSTADDRESS$ -C proc_smsd
```

Novell

MRTGEXT was originally written as an NLM for Novell Netware to obtain values used with the widely known MRTG (predecessor of cacti, which is the base of OP5 Statistics), but it can also be used to poll values from op5 Monitor.

Installing Novell MRTGEXT

To install this extension, simply copy the MRTGEXT.NLM to each NetWare server's SYS:SYSTEM directory that you wish to gather statistics from. Then edit the server's AUTOEXEC.NCF to “LOAD MRTGEXT” so it will load each time the server is restarted.

The MRTGEXT.NLM has three command line switches available:

- `-port=<port>`
  will change the port that MRTGEXT listens on for statistic requests. By default, MRTGEXT will use port `9999`. For example, to have MRTGEXT use port `1023`, add `-port=1023` to the load line. If you change the port number on the command line, be sure to modify the perl script as well.
- `-debug`
  will enable some debugging output to the System Console screen. This is only really useful when you are first configuring the extension.
- `-mla=<license>`
  For those with an MLA license (mostly for NetWare 5), the MRTGEXT.NLM currently can not obtain a valid value for the server license count. Using this option will tell the MRTGEXT.NLM the license count max to report. This is important if you use the NWEXTCFG.PL to create configuration files or if you use the servstat.pl script. For example, if you have a NetWare 5 MLA license and you really only have a 100 user server, then you would add `-mla=100` to your load command line.

More information

For more information please read here:

http://download.op5.com/agents/novell/1.46b/readme.txt

Windows SyslogAgent

op5 SyslogAgent runs as a service under

- Windows 2000
- Windows XP
- Windows 2003
- Windows 2008.

It formats all types of Windows Eventlog entries into syslog format and sends them to a syslog host (The op5 Monitor server or the op5 LogServer). The agent can also forward plaintext log-files.
Introduction

The entries in the Event log are sent to the op5 Logserver or op5 Monitor server. Text based application logs are also supported.

The op5 SyslogAgent is a repackaged version of the Datagram SyslogAgent, which initially is a bug fixed version of Sabre Net's old NT_Syslog. The op5 SyslogAgent is licenced as GPL software.

Installation

The op5 SyslogAgent installation package consists of an msi installer. To install simply double click the installation msi file and follow the on-screen instructions.

By default the op5 SyslogAgent will be installed in an op5 subdirectory to the program files folder. Usually:

```
C:\Program Files\op5\SyslogAgent\
```

You will also have the possibility to choose if you want to create start-menu, desktop and quick launch shortcuts to the SyslogAgent-configuration tool.

After the installation is completed you will be asked if you want to start SyslogAgentConfig. If you don't do this the agent won't be configured and cannot be started. If you choose to start the

SyslogAgent configuration continue to the section.

Upgrading

If a prior version of the SyslogAgent is installed it should, to avoid problems, be stopped and

uninstalled as a service and then uninstalled. Stopping and uninstalling the service can be done

from the SyslogAgent Configuration tool. Follow these steps to stop and uninstall the

SyslogAgent service:

1. Start the SyslogAgent Configuration tool
2. Press the “Stop”-button (see Fig 3. in the section Configuration)
3. Press the “Uninstall”-button

After the service have been stopped and uninstalled you should uninstall the previous version of

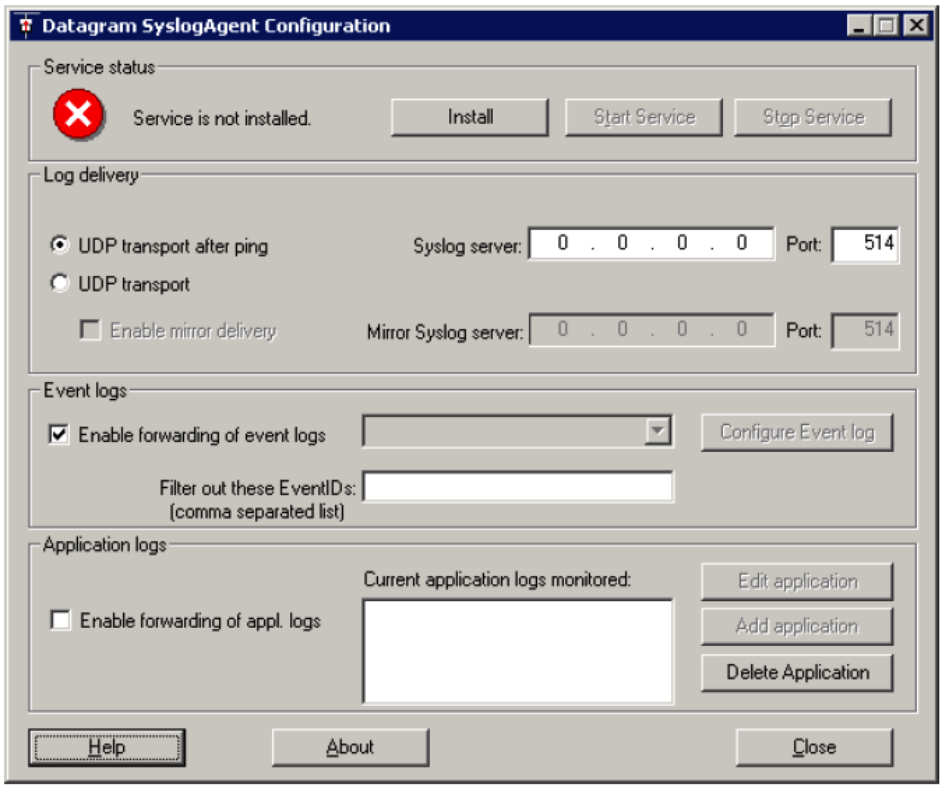
the SyslogAgent from “Add/Remove software” on the windows control panel.

Now you can proceed with the installation of the new version as usual. Note that your previous

settings will be used directly when the installation is complete.

Configuration

When the configuration tool is started the following window should be displayed:



Configuring the elementary functions

To configure the elementary functions and start the SyslogAgent started follow the following steps:

1 Press **Install**, this will install the SyslogAgent as a service.

2 Enter the IP address in the field **Syslog Server:**. This IP should be the one to your op5 Logserver or op5 Monitor server.

3 Make sure the check box “**Enable forwarding of event logs**” is checked.

4 Press **Start Service**.

Your SyslogAgent is now configured and should be sending logs to your op5 Logserver or op5

Monitor server.

Monitoring objects configuration

About Monitoring objects configuration

This chapter covers the following topics:

Introduction

There are two ways of changing the configuration of the op5 Monitor:

- Editing the configuration files in `/opt/monitor/etc`.
- Using the web UI **op5 Monitor configuration tool**.

In this chapter we will take a look at how the **op5 Monitor Configuration tool**, from now on called only **Configure**, is used.

Workflow

All configuration in op5 Monitor is saved in configuration files (text files) in `/opt/monitor/etc/`. The Configure works with a database and this makes it possible to do any changes in the configuration without saving it before you are satisfied.

The table below describes the workflow.

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>1</td><td>Configure opens and the configuration files are compared to the data in the database.
<table>
<tr><th>If</th><th>then</th><th>else</th></tr>
<tr><td>The configuration files are newer than the last change of the database</td><td>import the configuration files into the Configure database.</td><td>Do nothing besides open up the Configure.</td></tr>
</table></td></tr>
<tr><td>2</td><td>Edit the configuration</td></tr>
<tr><td>3</td><td>Save the changes to the Configure database by clicking Apply on the object you just added/changed.</td></tr>
<tr><td>4</td><td>When you are done with editing the configuration save the Configure database to the configuration files by clicking Save.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>5</td><td>A preflight check is made on the configuration before it is exported to the configuration files.
<table>
<tr><th>If</th><th>then</th><th>else</th></tr>
<tr><td>the preflight check founds any error</td><td>an error message is displayed and nothing will be exported</td><td>the configuration in the Configure database is exported and op5 Monitor is reloaded</td></tr>
</table></td></tr>
</table>

The basics

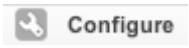
In The basics section we will take a look at the basic step you need to know about when working with **Configure**.

Start working

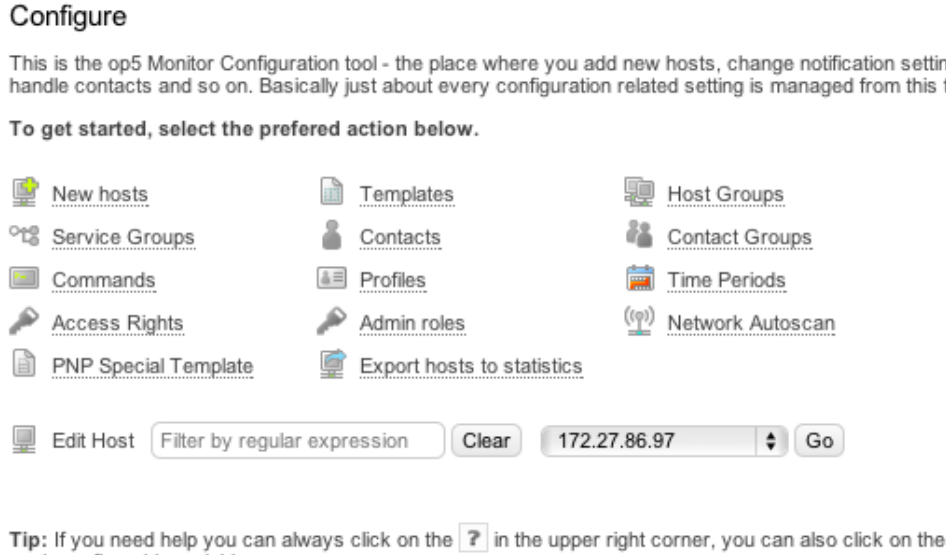
There are many ways to jump in to Configure and start working with the configuration of op5 Monitor.

To start working in Configure

Click Configure in the main menu



This will take you to the main menu of Configure.



or

Click the Configure icon on any object in the monitoring part of op5 Monitor

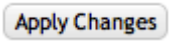
This will take you directly to the configuration part for the object you clicked on.

Applying changes

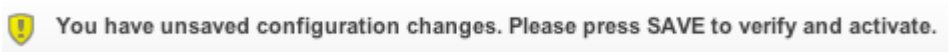
When you have made any changes to an object you have to apply it to the Configure database.

To apply the new configuration to the database

Click Apply at the bottom of the page



As soon as the data has been saved you will get the following warning telling you there is unsaved data in the Configure database.



Continue work until your work is done for this time.

Save the changes

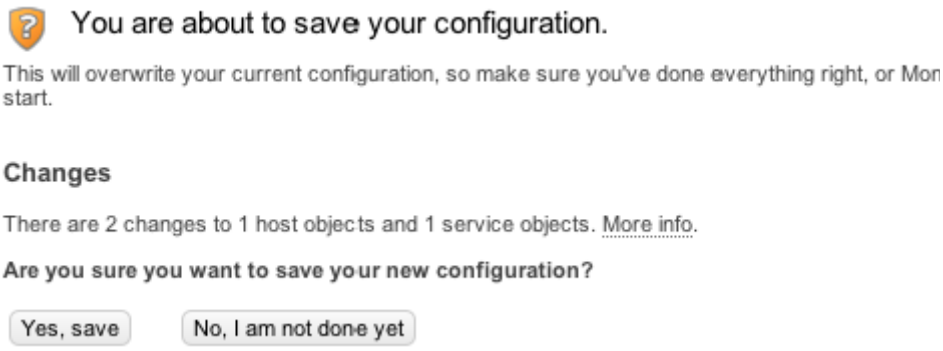
When you have finished working and consider your new configuration is ready to be used by op5 monitor you need to save the changes in the Configure database to the configuration files.

This will also make op5 Monitor start using the new configuration.

To save the changes and reload op5 Monitor

Click save icon at the top of the page.

Before the configuration is saved to disk, you have the opportunity to review the changes.



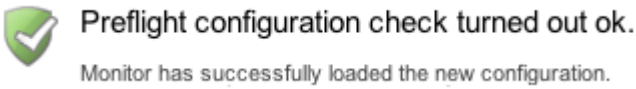
To view what changes that will be written to disk click on ‘More info’.

If you are another users is doing changes on the same objects that you have access to you will save the other users changes as well. The other users changes will be shown under ‘More info’ as well.

In the screenshot below you will see an example where we created a new host group and jsmith at the same time added a new host.

When done click 'Yes, save' to write all the changes to disk.

Now the preflight check is preformed and the data is saved to the configuration files.



Note: if two users with the same permissions are editing the same host all configuration regarding the host or service will be saved.

Undo changes

Sometimes it might be handy to reset the configuration to the state it was in where you started to work in Configure. The only thing you have to do then is to undo your changes.

The undo function will only work as long as you do not have saved the data to the configuration files.

To undo the configuration changes.

Click undo icon at the top of the Configure page.

This will revert the your changes since the last successful preflight check.

Changes reverted.
You might want to do a complete reimport?

To undo all users changes click on complete reimport. This will re-read the configuration files and all changes will be reverted. If any changes were made directly into the configurations files these changes will now be loaded in to the web configuration

Import forced
Configuration has been imported, overwriting database changes.

Historical Configuration Changes

Historical configuration changes can be used to track changes in the configuration. In the log you will find all changes in the configuration on objects that you have access to.

To access the historical configuration changes log, go to ‘Configure’ and click on the icon in the upper right corner.

Limited users will only see changes that are made to the hosts and services they are contacts for.

Full access users will see all changes.

User	Object type	Object	Action
monitor	hostgroup	DEMO_group	Created new object **DEMO_group**
jsmith	host	My_Server	Created new object **My_Server**
jsmith	service	PING on host My_Server	Created new object **My_Server;PING**
monitor	service	smtp on host WinServer	Created new object **WinServer;smtp**

Main objects

The configuration is based on objects. There are several types of objects, each one defining different things in the monitoring process.

Each object consists of a object name and a couple of variables that needs to be configured.

For example on a host object you configure

- host name
- address
- notifications
- active checks
- etc.

In Configure you can

- add new objects
- modify existing objects
- remove existing objects.

A lot of objects can be cross referenced in the configuration and Configure helps you with this to.

In most of the listings you will find a small text field called **Filter by regular expression**. Use this to filter out the content you are interested in when viewing the different lists.

Required directives

All objects have a list of directives that are required when adding a new object. The other directives can be left out. They will then get the op5 Monitor defaults value.

This does not mean you have to set every directive for every object. One solution is called templates. They make it a lot easier to manage a large set of objects. Read more about templates in *Using templates* on page 45.

Hosts

Hosts are one of the central objects in the monitoring logic. Important attributes of hosts are as follows:

- Hosts are usually physical or virtual devices on your network (servers, workstations, routers, switches, printers, etc) but it could be practically anything you can reach and monitor from the op5 Monitor server.
- Hosts have an address of some kind, IP address or host name.

- Hosts does not need a service directly associated to them, the services can be inherited from a hostgroup. A host can also exist without services.
- Hosts can have parent/child relationships with other hosts, often representing real-world network connections, which is used in the network reachability logic.

Required directives

The following directives are required for a host object.

- host_name
- alias
- address
- max_check_attempts
- check_period
- contacts
- contact_groups
- notification_interval
- notification_period

The table below describes the required directives for the host object

Directive	Type	Description
`host_name`	string	This is the id of the object. I may not contain any space in the value.
`alias`	string	A more describing name for the object.
`address`	string	The address the host is reached by, preferably an IP address to make sure the host is reachable even if the DNS is down.
`max_check_attempts`	integer	Is used to define the number of times op5 Monitor will retry checking the host if it returns any kind of problem state. Setting this value to 1 will cause op5 Monitor alert directly without any retry.
`check_period`	time_period	During this period the host is checked. It can be any time period defined in op5 Monitor.

Directive	Type	Description
`contacts`	contact	Single contacts used to send notifications to and gives access to this host for users who do not have access to all hosts.
`contact_groups`	contact_group	Contact groups used to send notifications to and gives access to this host for users who do not have access to all hosts.
`notification_interval`	integer	Number of minutes between renotifications. Set this to 0 if you only want to have one notification sent out.
`notification_period`	time_period	During this period the notifications are sent out if any alerts are created. It can be any time period defined in op5 Monitor.

Services

A service can be practically any thing that you can measure and monitor on a host. It is almost only your imagination and programming skills that sets the limit for what you can monitor with a service.

A service

- must be connected to a host
- can check things by tcp, agents, snmp etc.
- use a check command (*Commands* on page 40) to communicate with the plugin (*Plugins* on page 40) that gets all the data.

Required directives

The following directives are required for a service object.

- host_name
- service_description
- check_command
- max_check_attempts
- check_interval
- retry_interval
- check_period

- notification_interval
- notification_period
- contacts
- contact_groups

The table below describes the required directives for the host object

Directive	Type	Description
`host_name`	host_name object	The host the service is connected to.
`service_description`	string	This is the id of the object. It must be unique on a host but may be reused on other hosts.
`check_command`	command object	This is the short name of the command that is executed during service checks.
`max_check_attempts`	integer	Is used to define the number of times op5 Monitor will retry checking the host if it returns any kind of problem state. Setting this value to 1 will cause op5 Monitor alert directly without any retry.
`check_interval`	integer	The number of minutes between normal service checks.
`retry_interval`	integer	The number of minutes between retry checks when a service has gone into a problem state before the state becomes hard.
`check_period`	time_period	During this period the service is checked. It can be any time period defined in op5 Monitor.
`contacts`	contact	Single contacts used to send notifications to and gives access to this host for users who do not have access to all hosts.
`contact_groups`	contact_group	Contact groups used to send notifications to and gives access to this host for users who do not have access to all hosts.

Directive	Type	Description
notification_interval	integer	Number of minutes between renotifications. Set this to 0 if you only want to have one notification sent out.
notification_period	time_period	During this period the notifications are sent out if any alerts are created. It can be any time period defined in op5 Monitor.

Contacts

A contact is used for two purposes:

- to send notifications to
- permissions to view a objects in the monitoring part of op5 Monitor.

A contact is not the same as the login account given access rights to the system.

Required directives

The following directives are required for a service object.

- contact_name
- host_notifications_enabled
- service_notifications_enabled
- host_notification_period
- service_notification_period
- host_notification_options
- service_notification_options
- host_notification_commands
- service_notification_commands

The table below describes the required directives for the host object

Directive	Type	Description
contact_name	string	The id of the contact object.

Directive	Type	Description
host_notifications_enabled	yes/no	Used to determine whether or not the contact will receive notifications about host problems and recoveries.
service_notifications_enabled	yes/no	Used to determine whether or not the contact will receive notifications about service problems and recoveries.
host_notification_period	time_period object	The time period when the contact will receive any host notifications.
service_notification_period	time_period object	The time period when the contact will receive any service notifications.
host_notification_options	Down, Unreachable, Recovery, Flapping start and stop, Scheduled downtime start and stop	Used to set what type of host notifications the contact shall receive.
service_notification_options	Critical, Warning, Unknown, Recovery, Flapping start and stop, Scheduled downtime start and stop	Used to set what type of service notifications the contact shall receive.
host_notification_commands	command object	The command used to send the host notifications
service_notification_commands	command object	The command used to send the service notifications.

Directive	Type	Description
notification_period	time_period	During this period the notifications are sent out if any alerts are created. It can be any time period defined in op5 Monitor.

Access rights

Access rights is actually user accounts that makes it possible to login to the op5 Monitor GUI.

The access rights does not have any thing to do with notifications or the permissions of viewing objects in op5 Monitor.

Access rights can be connected to a contact by giving the username the same name as the id (contact_name) of a contact.

Required directives

The following directives are required for a access rights object.

- username
- password

The table below describes the required directives for the host object.

Directive	Type	Description
username	string	The username is the id of the access rights and also used as login username.
password	string	The password is used for the login.

Access rights in detail

The table below gives you a description of the settings of an access right object.

Directive	Description
authorized_for_system_information	Gives the user access to the system / process information.

Directive	Description
authorized_for_configuration_information	Gives the user access to view and change configuration.
authorized_for_system_commands	Gives the user access to issuing commands in the web gui. With commands you can control certain functions in op5 Monitor, for example: enable/disable notifications, scheduled downtime, acknowledge problems etc.
authorized_for_all_services	Gives the user access to view all services, se Customizing views below for more information.
authorized_for_all_hosts	Gives the user access to view all hosts, se Customizing views below for more information.
authorized_for_all_service_commands	Gives the user access to issue commands for all services, se Customizing views below for more information.
authorized_for_all_host_commands	Gives the user access to issue commands for all hosts, se Customizing views below for more information.

Recommended settings

Recommended settings for

- an administrator would be to check all boxes
- help desk staff it could be
  - authorized_for_system_information
  - authorized_for_system_commands so they can acknowledge problems but not change the configuration.

Admin Roles

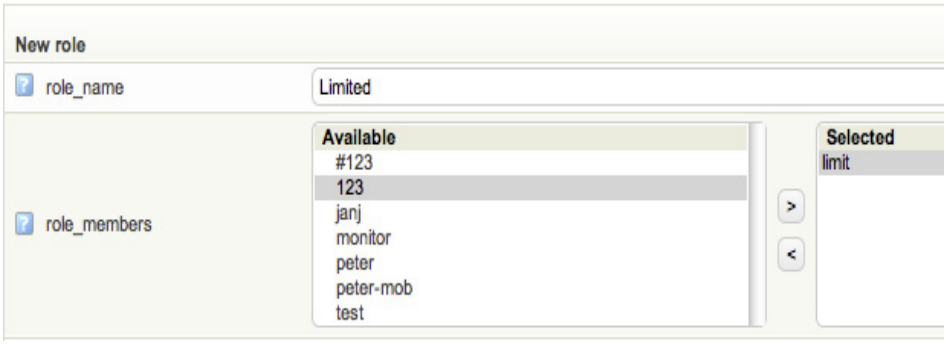
Admin roles make it possible to create a role with more fine-grained permissions than with Access Rights

When combining several different permissions, the role's members will only get access to the objects that are allowed according to all of the permissions.

If at least one permission is selected, and none of the permissions is either "Allow all" or "Disallow all", then an "Allow all" will be implied. That is, if you choose only "Disallow creating new objects", that gives the role permission to edit all existing objects.

Leaving a permission empty means it will fall back to the permission in the "default" role, and that grants full access.

The option "Only contacts can access" roughly means that whatever is visible in op5 Monitor will be visible in the configuration.



A role can be used to limit access to the following configuration objects.

Variable	Function
Role Members	Assign members to the role
Host	Setup how the users are going to be allowed to modify host objects
Hostgroup	Setup how the users are going to be allowed to modify hostgroup objects
Service	Set up how the user are going to be allowed to modify services
Contact	Setup how the users are going to be allowed to modify contact objects
Contact Group	Setup how the users are going to be allowed to modify contactgroup objects
Time Period	Setup how the users are going to be allowed to modify timeperiod objects
Command	Setup how the users are going to be allowed to modify command objects
Service Dependency	Setup how the users are going to be allowed to modify servicedependency objects
Service Escalation	Setup how the users are going to be allowed to modify serviceescalation objects
Service Group	Setup how the users are going to be allowed to modify servicegroup objects

Variable	Function
Host Dependency	Setup how the users are going to be allowed to modify hostdependency objects
Host Escalation	Setup how the users are going to be allowed to modify hostescalation objects

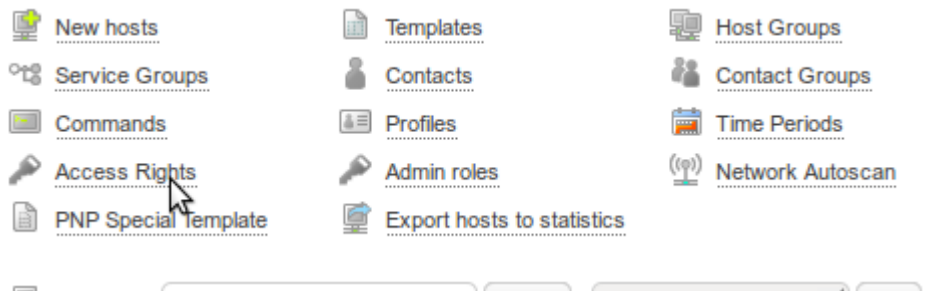
These permissions have no effect if a user is ‘authorized_for_all_hosts’ under Access Rights.

Admin Role Example

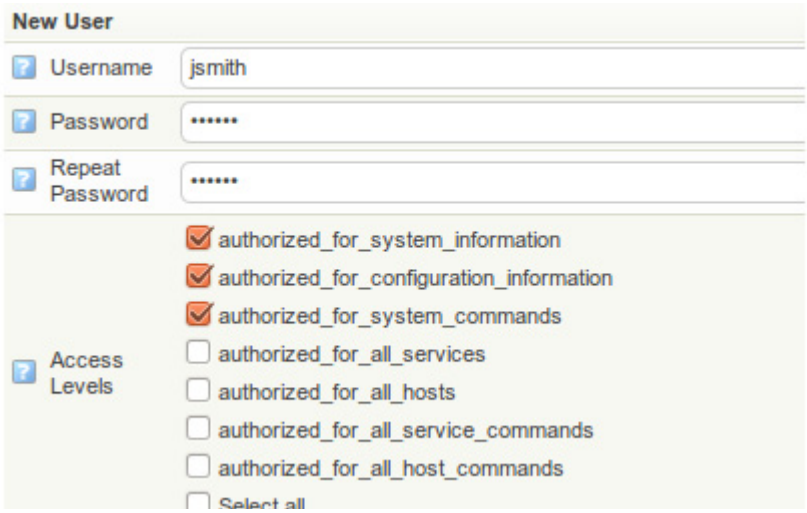
In this example we will create a user that has the ability to see all hosts in the assigned hostgroup, and can edit services in this hostgroup, in this case Mysql-srv1.

Prerequisites:
You have a contact that we call jsmith that is member of contact group "Database" that is assigned to a number of hosts in a hostgroup.

1 Go to: Configure -> Access Rights

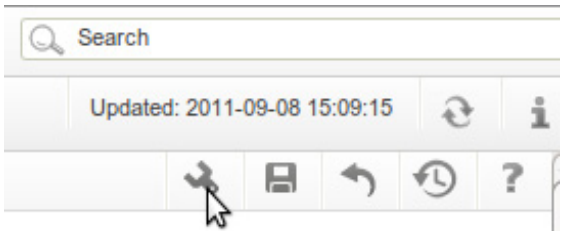


2 This contact has the access rights: authorized_for_system_information, authorized_for_configuration_information and authorized_for_system_commands.

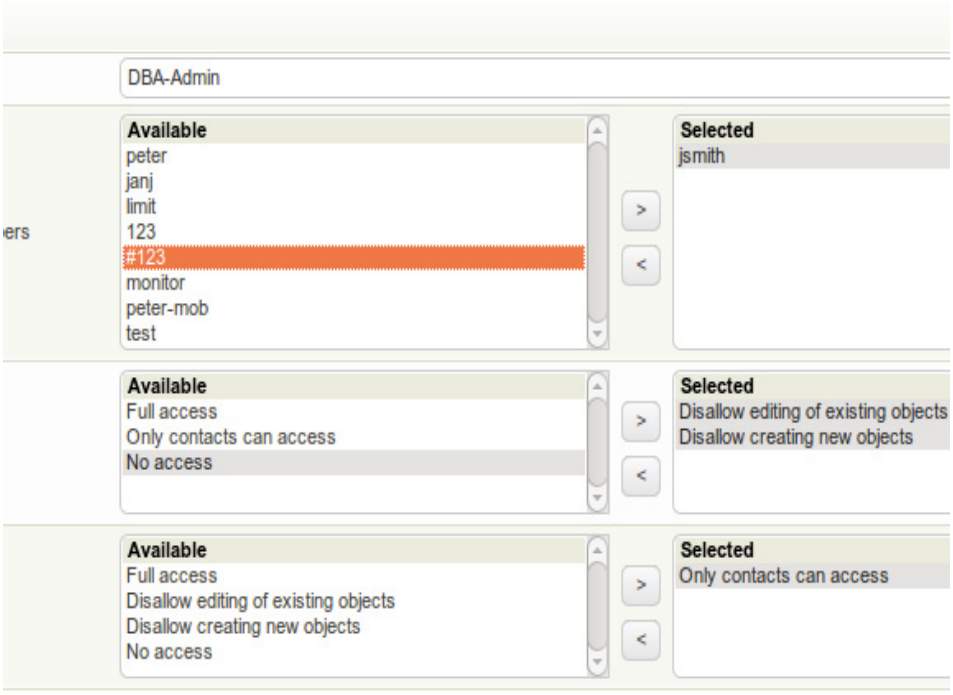


3 Click "Apply Changes"

4 Then go to "Configure"



5 Click "Admin roles"
Fill out the configuration options in all fields.



Note: If you leave a field empty, "Full Access" will be implied.

When combining several different permissions, the role's members will only get access to the objects that are allowed according to all of the permissions.

If at least one permission is selected, and none of the permissions is either "Allow all" or "Disallow all", then an "Allow all" will be implied.
That is, if you choose only "Disallow creating new objects", that gives the role permission to edit all existing objects.

Leaving a permission empty means it will fall back to the permission in the "default" role.

"Only contacts can access" roughly means that whatever is visible in Ninja will be visible here.

These permissions have no effect if a user is authorized_for_all_hosts in "Access Rights".

Example

In this case we will create a role called DB-Admin that only will have access to the hostgroup "Databases".

This role can edit services on hosts in this hostgroup, add new services to the hosts in this hostgroup and create and edit services, but not create new hosts or se other hosts than “jsmith” is assigned to.

Select the following options:

Role-name	DB-Admin
Role_members	jsmith
host	Disallow editing of existing objects, Disallow creating new objects, Only contacts can access
hostgroup	Only contacts can access
service	Only contacts can access
contact	No access
contactgroup	No access
timeperiod	Full access
command	Full access
servicedependency	No access
serviceescalation	No access
servicegroup	Only contacts can access
hostdependency	No access
hostescalation	No access
FILE	No access
access_rights	No access

Then you logout as admin and login with the user "jsmith".

Now "jsmith" can only access his hostgroup and change the services associated in the hostgroup "Databases".

Time periods

Time periods is time defining objects that span over a week. You can define included time for each day of the week in the time period definition.

You can also:

- use already defined time periods as excludes
- add exceptions based on dates and ranges of days

The time period objects are used at many places in the configuration. Most noticeably are in the contact objects where the time periods defines when notifications should be sent out.

You can also use time periods to define when a service or a host should be monitored or when you are creating availability reports.

A time period in detail

The following tables describes the directives of a time period and how to use them.

The table below describes the first part of directives of a time period.

Directive/option	Description
`timeperiod_name`	short name of the time period
`alias`	descriptive name of the time period
`Monday to Sunday`	which time to include for each day. you can define multiple times by separating them with comma. Example 00:00-01:00,03:00-06:00
`Exception type`	Specify what type of exception you want to use; Date or Day

Depending on what kind of exception type you have chosen you will get different settings choices. The two lists below describes them all.

The table below describes the exception part of a time period.

Directive/option	Description
`exclude`	Other predefined time period definitions that should be excluded from this time period.
`Exception type`	Specify what type of exception you want to use; Date or Day

The table below describes exception by **Date**:

Directive/option	Description
`Interval`	Choose Single ate or Date range
`Date`	Choose the date that is supposed to be used in this Exception.
`From date`	If you chosen date range you will here set the start date To date.
`To date`	If you chosen date range you will here set the end date.

Directive/option	Description
`Frequency`	How often the exception is repeated. Valid values are positive integers greater than one. E.g: • Date range "2008-01-01 - 2008-12-31 / 5" means every fifth day of 2008. • Day range "1 monday march - 3 sunday may / 3" means every third day between the first monday and the third sunday every month. • Date range "2008-06-01 / 14" means every 14th day from first of june 2008. Note that this exception has no end.
`Hours`	Which time to include for this exception. You can define multiple times by separating them with comma. Example: 00:00-01:00,03:00-06:00

The table below describes exception by **Day**:

Directive/option	Description
`Interval`	Choose Single day or a Day range
`Weekday`	Choose the weekday that is supposed to be used in this Exception.
`From weekday`	If you chosen Day range you will here set the start day.
`To weekday`	If you chosen Day range you will here set the end day.
`Frequency`	How often the exception is repeated. Valid values are positive integers greater than one. E.g: • Date range "2008-01-01 - 2008-12-31 / 5" means every fifth day of 2008. • Day range "1 monday march - 3 sunday may / 3" means every third day between the first monday and the third sunday every month. • Date range "2008-06-01 / 14" means every 14th day from first of june 2008. Note that this exception has no end.
`Hours`	Which time to include for this exception. You can define multiple times by separating them with comma. Example: 00:00-01:00,03:00-06:00

Commands

A command is exactly what it sounds like. It can use macros and arguments. Mostly they are used with services but they can actually be used as

- service or host check command
- notification command
- event handler
- obsession.

Directives

A command has got only two directives

- command_name
- command_line

Directive	Description
`command_name`	This is the id of the command and also the name shown in Configure.
`command_line`	is the actual command line used by the services, notifications, event handlers and obsession.

Plugins

Plugins are compiled executable or scripts that can be run from a command line to check the status or a host or service.

There are many plugins included in the op5 Monitor software. A list of the plugins can be found in the **list-of-plugins** at the support section at www.op5.com.

If you are looking for a plugin not found in op5 Monitor by default there are a bunch of other places to look

- contact op5 for a specific development
- www.op5.org
- exchange.nagios.org

You can use any plugin written for Nagios but you might need to modify them a bit before they work in the op5 Monitor environment.

Groups

The groups in op5 Monitor is used to group objects of the same type. There are three types of groups in op5 Monitor

- host groups
- service groups
- contact groups

They are all good to use to get things a bit more organized and they have also special functions op5 Monitor.

The following subsections will give you a brief description about how they can be used.

You may not have any empty groups in op5 Monitor so when you create a new group, no matter what type it is, you should have at least one object to add to the group.

Host groups

Host groups can be used group hosts together in any way you like.

- A host can be connected to any number of hosts.
- A host group can be connected to an other host group.

There are a few host groups included in the initial setup of op5 Monitor but you can create your own matching your own needs.

There are a infinite ways of using host groups and here are a couple of examples.

Grouping hosts by

- geographic placements
- what company they belongs to
- who owns the hosts
- who should be able to see the hosts in the group
- function or operating system.

The list can be long.

Services on Host groups

A host group can contain service checks. These service checks will be inherited on all hosts connected to the host group.

A service on a host group work in the same way as a service for a host.

To add a service to a host group go to ‘Configure’ and ‘Host Groups’. Choose the host group you want to add services to then select ‘Services for hostgroup’

For example a windows servers host group could contain the checks that are common for all windows servers. By doing this you will only need to change command arguments on the service in the host group instead of changing the arguments on all windows host.

If you add new checks to the service group all hosts in the host group will get the new service once you save your configuration.

If a host group service and a host service should get the same name, the host group service will be used, the host service will still be visible in the configuration and if the host is lifted out from the host group the host service will become active.

Nested host groups

Host groups can be connected to each other.

When nesting host groups together the services on host groups also will inherited to the nested host group. This only work one way.

For example:
Host group A has service X and host 1 is a member of host group A
Host group B has service Y and host 2 is a member of host group B
If host group B is added as a member of host group A then host 1 will get service Y but host 2 will not get service X.

A good way to use this feature is to have i.e a Windows host group and then a MSSQL host group. When adding the Windows host group as a member to the MSSQL host group the hosts added to MSSQL will get both the service checks that are standard for all Windows host and the default MSSQL service checks.

Service groups

The service groups are used to group services together in the same way as for host groups. On the other hand there is almost no useful at all to for example group service groups by geographic placements.

One good way to use service groups is to create groups containing services needed for a service you deliver to your customers.

Example 1 *An email service group*

Let us take a simplified email service and show how the service groups can be used.

To be able to deliver an email service to our customers the following services need to be working:

- *DNS*
- *SMTP*
- *IMAP / POP3*
- *WAN Connection*
- *File Storage*

We take al those services and place them in a service group called Customer email.

If we get a problem with any of the services in the Customer email group we can easily see that the whole email service has got a problem.

The service group in the example above is perfect to use in Service Level Agreement reports (SLA in the op5 Monitor user manual) to make sure we deliver the service as we promised.

Contact groups

Contact groups are mainly used to setup where to send service and host notifications. It can also be used to setup permissions about who should be able to see what object in the op5 Monitor GUI.

The members of a contact group associated with a certain host and/or service are the one that will get all notifications for that object.

A Contact group can be populated with a contact or another contact group.

Permission to host and services

If a user does not has the access rights to see all hosts that user need to have a contact connected to the contact group associated with the host or service the user should be able to see.

Show partial hostgroups

If an unprivileged user is not a contact for all hosts in a hostgroup, he will not be able to see the host group in the "Hostgroup summary/overview/grid" views.

To enable viewing of partial host group edit follow these steps logged in as root:

1. Create and edit the file /opt/monitor/op5/ninja/application/config/custom/groups.php with your favorite editor.
2. Put the following into the file:

```
<?php defined('SYSPATH') OR die('No direct access allowed.');
$config['see_partial_hostgroups'] = true;
```

3 Save the file.

Using templates

Even though Configure makes it easy for you to add and change the configuration of op5 Monitor it is still a lot of things to edit and tweak. To make the software even more easy to use templates have been built in.

There are three types of templates to use:

- host templates
- service templates
- contact templates

op5 Monitor comes with a couple of predefined templates for each object type described above. They are just there to be examples and you should really create your own.

How they work

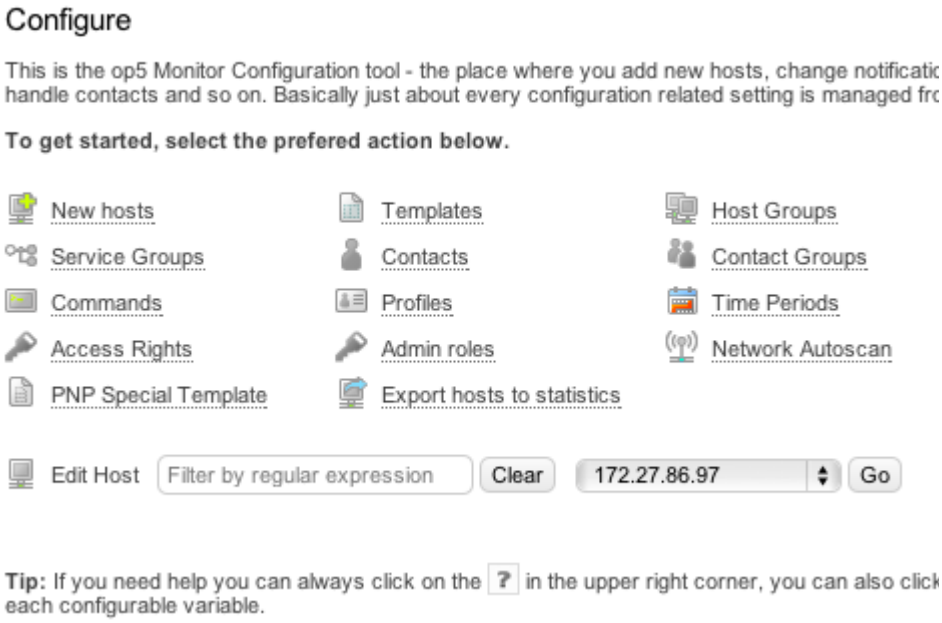
- Any directive set in a template will be used in the objects using the template. But if you set a directive explicit on an object that value will override the templates.
- Any directive not set in neither a template or directly on the object will have the op5 Monitor default value.
- If you change any value on a directive in a template it will only be valid on the objects where the same directive is not set explicit.

Managing objects

Now let us be a bit more hands on. In this section we will take a look at how to add/edit/delete objects using the Configure.

There are sometimes many ways to do things in op5 Monitor but we will only show a few examples.

In the subsections to Managing objects we will assume that you start from the main page of Configure.



Before you start

Add new

Every time you comes to a page where you can handle an object you will have the **Add new...** dialog ready for you to add a new object.

Configuration files

Every object is placed in a configuration file. You may change what file the object is placed in at the bottom of every configuration page. This is normally not necessary and only used in special cases.

Help

In the guides we will only describe the directive that are differ from the default value. Click the help icon

Templates

Because handling templates is the same for all kind of templates, only the directives differ, we will only add a template in *Contacts* on page 47.

Contacts

Adding a contact template

Before we start to add any new contacts we will create a contact template to use with the contact in the next section. In this guide we only describes the directive we will not use the default value in.

To add a contact template

1 Click **Templates** on the main page.

2 Click **Contact templates**.

3 Give the contact template a name

4 Change **can_submit_commands** to yes. [1]

5 Click **Apply changes**.

6 Click **Save**.

Adding a contact

To add a contact

1 Click Contacts on the main page.

1.This gives this the user connected to this contact the possibility to execute commands like acknowledge problems etc.

2 Use the template on call template we created in *Adding a contact template* on page 47.

3 Type in a contact_name

4 Type in an alias

5 Type in the email address

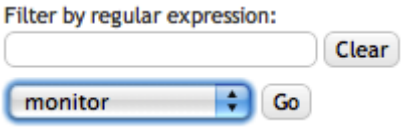
6 Click Apply changes.

Modify a contact

To modify a contact

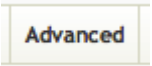
1 Click Contacts on the main page.

2 Choose the contact you like to modify in the drop down list.



3 Click **Go**.

4 In the view you will get only directives differ from the template will be shown. To change the other directives click **Advanced**.



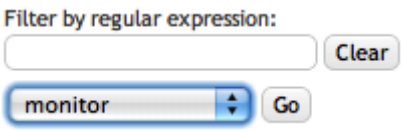
5 Make your modifications and click **Apply changes**.

6 Click **Save**.

Delete a contact

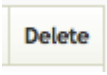
1 Click Contacts on the main page.

2 Choose the contact you like to modify in the drop down list.



3 Click **Go**.

4 Click **Delete**.



5 Click **Save**.

Hosts

There are many ways to add a host. A host can be added by

- the **new host** wizard
- a **network scan**
- cloning of a host
- using a profile.

In this guide we only describes the directive we will not use the default value in.

Adding a host with new host wizard

To add a new host using the new host wizard - Part 1

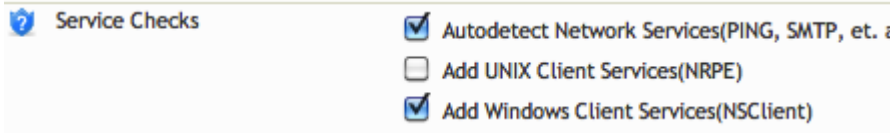
1 Click **New host** on the main page.

2 Type in a host_name.

3 Type in an alias.

4 Type in the address to the host, IP address is mostly the best choice.

5 We assume this is a Microsoft windows server and that NSClient++ has been installed. Check for the following service checks.



6 Click host logo to set the icon that will be displayed for this host in lists and maps.

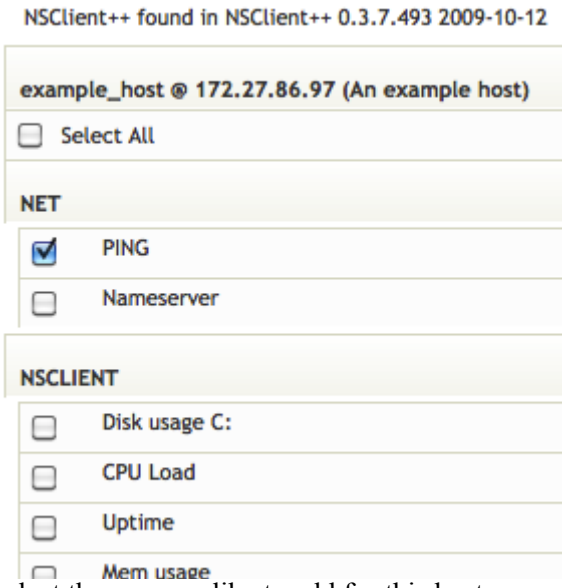
A list looking like this will be displayed. Click the icon you like to use.

7 Click **Scan hosts for services**.

To add a new host using the new host wizard - Part 2

1 Leave the initial settings [1] as it is and scroll down to the services.

2 The scan has found out that NSClient++ is installed plus two other services that can be added to this host. Check Select All to add all services found or



select the one you like to add for this host.

3 Click **Continue to step 3**.

1.All new services will inherit the Initial Service Settings. If you choose not to enter a value for one or more required variable, those variables must be set in the selected template.

4 Now either click the host or service links or click **Save**.

Adding hosts with network scan

Network ranges can be specified in a very free form. Each of the four parts of the IP-address may contain any combination of comma-separated numbers, 'from-to' ranges and single numbers, as such: 10.1,2.0,4-10.1-50.

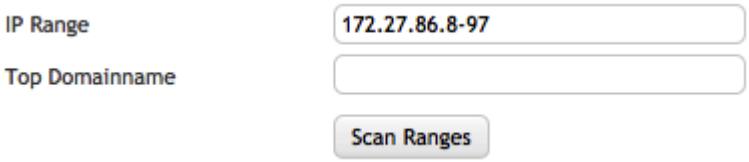
You can specify multiple ranges, separated by spaces, if you like.

To add hosts with network scan

1 Click **New host** on the main page.

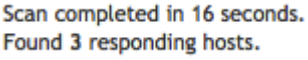
2 Click **Network scan**.

3 Fill in the desired network range. We will scan for hosts in the range from 172.27.86.8 - 172.27.86.97.



4 Click **Scan Ranges**.

5 In this case we found [1] three hosts.



6 Repeat *To add a new host using the new host wizard - Part 1* on page 49 for each host, except for the last step. If here is one or more host you do not like to add choose **No** in **Add this host?**
When you are finished click **Scan hosts for services**.

7 Repeat *To add a new host using the new host wizard - Part 2* on page 50 for each host, except for the last step.
When you are finished click **Continue to step 3**

8 Click **Save**.

1.Only hosts that aren't previously configured will be listed

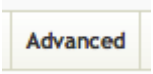
Modifying a host

To modify a host

1 On the start page choose the host you like to modify in the drop down list.

2 Click **Go**.

3 In the view you will get only directives differ from the template will be shown. To change the other directives click **Advanced**.



4 Make your modifications and click **Apply changes**.

5 Click **Save**.

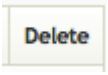
Deleting a host

To modify a host

1 On the start page choose the host you like to delete in the drop down list.

2 Click **Go**.

3 Click **Delete**.



4 Click **Delete all affected objects**.

5 Click **Save**.

Renaming a host

When renaming a host in the web GUI it will only rename the host and will not rename the host name.

To rename the host name in log-files as well a script has to be run manually. The script will rename the host in log-files. If this is not done the host will lose all it's alert history.

To run the script logon to the op5 monitor via SSH as root user and execute the following command:

```
# mon stop; /opt/monitor/op5/merlin/rename --rename-all; mon start
```

If there is a lot of history this script can take a while to execute and during this time the op5 monitor service will not be running.

Note that this does not yet work on schedule downtime objects. If a host is renamed that has a scheduled downtime the scheduled downtime will be lost.

Network autoscan

It might get handy to let op5 Monitor scan and notify you if there are any new hosts on a particular network range.

The network autoscan function will

- scan certain range for new hosts
- notify you when new are found
- be executed every night by cron on the op5 Monitor server.

No host will be automatically added. The network autoscan function will only find the hosts for you.

Adding a new autoscan configuration

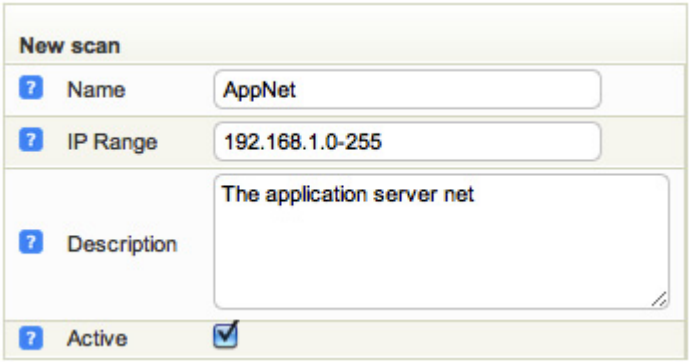
You may add as many autoscan configuration as you wish. When adding a your network range you may use the same syntax as when you manually scans a network from the Add new host wizard.

To add a new autoscan configuration

1 Click **Configure** in the main menu.

2 Click Network Autoscan.

3 Fill in the **New scan** form



- **Name**: The identifier of this autoscan configuration
- **IP Range**: In this case a complete C net.

- **Description**
- **Activate**: Make this autoscan configuration active and in use.

4 Click **Save**.

Adding a host to blacklist

In certain ranges you are scanning with the network autoscan there might be hosts you do not want to include in the result. Then you should add that host or hosts to the blacklist.

To add a host to the blacklist

1 Click **Configure** in the main menu.

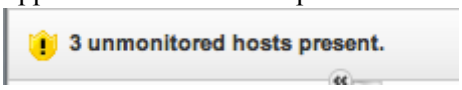
2 Click Network Autoscan.

3 Add a host (IP address) in the **Host** field

4 Click **Add**.

The result

After the networks scan has been executed a small result will be shown in the upper left corner of the op5 Monitor GUI



To add the hosts that has been found you only need to click on the text to the right of the icon. You will then come to the Add new host wizard the same as when you have done a manual network scan.

Services

Services can be added in a few different ways in Configure. You may add a service by using

- **add service for this host**
- **scan host for network services**
- **scan host for snmp interfaces**
- cloning.

We will take a look at the **add service for this host.**

In this guide we only describes the directive we will not use the default value in.

The default service template will used.

Adding a service

To add a service using add service for this host

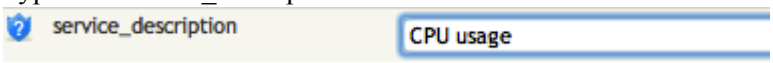
1 On the start page choose the host you like to add a new service to in the drop down list.

2 Click **Go**.

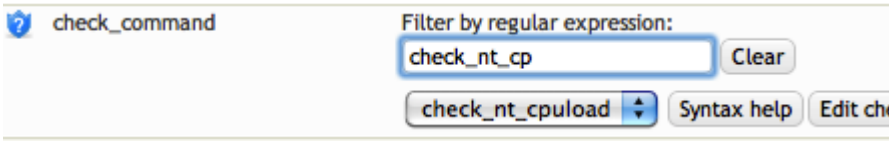
3 Click **Services for host...**.

The add new service pages is shown.

4 Type in a service_description.



5 We will use the check_nt_cpuload command for this service.
Type in as many chars you need in the filter by regular expression field until the command shows up.



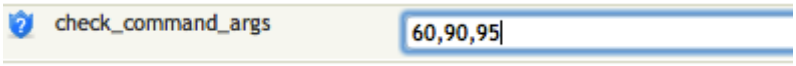
6 Click Syntax help to see what arguments are needed for this command.

```
Command line of selected check command:
/opt/plugins/check_nt -H $HOSTADDRESS$ -p 1248 -v CPULOAD -l$ARG1$
Plugin syntax:
This plugin collects data from the NSClient service running on a
Windows NT/2000/XP/2003 server.
Usage:check_nt -H host -v variable [-p port] [-w warning] [-c
critical][-l params] [-d SHOWALL] [-t timeout] [-T timeout_status]
Options:
 -h, --help
    Print detailed help screen
 -V, --version
    Print version information
Options:
 -H, --hostname=HOST
```

You can see that we have a macro called **$ARG1$**. This is the first, and in this case the only, argument we need to give to this command.

7 Click **Syntax help** again to hide the help text.

8 Type in the argument [1].



9 Click **Apply changes**.

10 Click **Save**.

Modifying a service

To modify a service

1 On the start page choose the host you like to modify a service on in the drop down list.

2 Click **Go**.

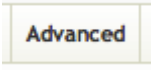
3 Click **Services for host ...** .

4 Choose the service you like to modify in the drop down list.

5 Click **Go**.

1.If more than one the shall be separated by a ! like this: argone!argtwo.

6 In the view you will get only directives differ from the template will be shown. To change the other directives click **Advanced**.



7 Make your modifications and click **Apply changes**.

8 Click **Save**.

Test this service

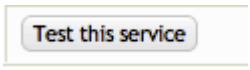
Test this service makes it possible for you to test the service you added or modified before you save the new configuration and reload monitor. This is a nice way to make sure the service works as it is supposed to.

In the guide below we will work with the service created in *Adding a service* on page 55.

To test a service

1 Pick up the service you like to test as it is done in *Modifying a service* on page 56.

2 Click **Test this service**, at the bottom of the page.



3 The output looks like the one below. If you get any errors it will be shown

```
Command: /usr/bin/asmonitor /opt/plugins/check_nt -H 172.27.86.97 -p
1248 -v CPULOAD -l60,90,95
Seen command (one argument per line):
  /opt/plugins/check_nt
  -H
  172.27.86.97
  -p
  1248
  -v
  CPULOAD
  -l60,90,95

Plugin output:
CPU Load 1% (60 min average) |   '60 min avg Load'=1%;90;95;0;100

Plugin return code: 0
```

here in the output

4 Click **Test this service** again to hide the output.

Deleting a service

To delete a service

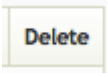
1 On the start page choose the host you like to delete a service from in the drop down list.

2 Click **Go**.

3 Click **Services for host ...** .

4 Choose the service you like to modify in the drop down list.

5 Click Delete.



6 Click **Save**.

Scanning host for network services

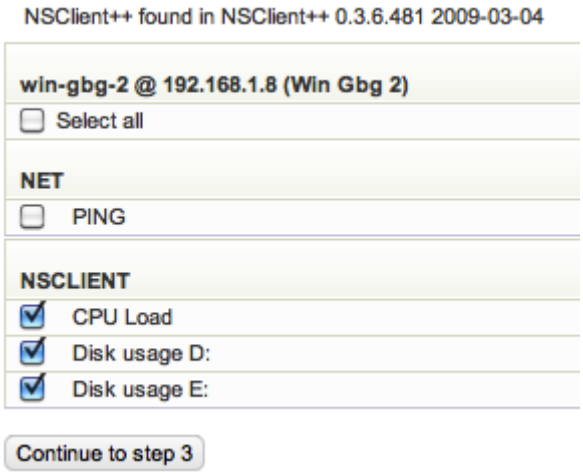
When you added your host (*Hosts* on page 49) you had the opportunity to add services found during the scan for network services. This scan function can also be reached afterwords.

To scan a host for network services

1 Open up the host, in **Configure**, you like to add new services on.

2 Click **Scan host for network services**.

3 Select the new services found and click Continue to step 3.



4 Click either the host or service link to go back to the place where you started.

5 Click **Save**.

Scanning a host for snmp interfaces

In many times when you are about to monitor a switch or a router you need to setup a lot of services. It is hard work and takes a lot of time to add them one by one.

Instead of adding all interface services one by one you should use the scan for snmp interfaces function.

To add snmp interfaces

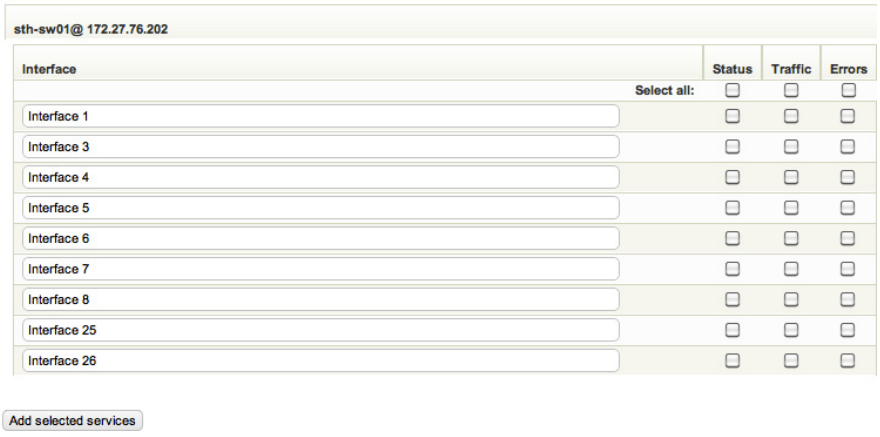
1 Open up the host, in **Configure**, you like to add new services on.

2 Click **Scan host for SNMP interfaces**.

3 Set the SNMP community.

4 Chose SNMP version.

5 Click **Scan host**.



6 Select the services you like to add.

7 Click either the host or the service link to get back.

8 Click **Add selected services**.

9 Click **Save**.

Scanning host for windows services

Adding a service that checks a windows services is many times harder than you think. You need to

- have access to the windows server
- know the exact name of the windows service

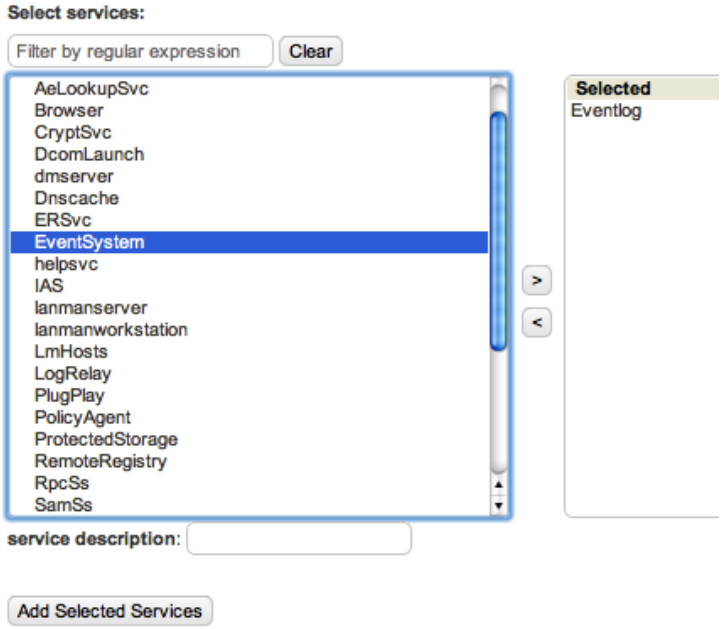
With op5 Monitor you do not need to do anything more than make sure the latest agent (NSClient++) is installed and follow the next few steps.

To add windows services

1 Open up the host, in **Configure**, you like to add new services on.

2 Click **Scan host for Windows Services**.

3 Select the Windows Services you like to add as a new service in op5 Monitor.



4 Give the new service a **Service description**.

5 Click **Add Selected Services**.

6 Click either the service link or the **Scan for more service** button.

7 Click **Save**.

Escalations

Escalations let you configure escalation of notifications for this host. The idea is that if you have a really important host you can send the first notification to the default contact group in order for them to solve the problem. If the problem is not solved in lets say 30 minutes you can send the notification to a broader range of contacts.

Host and service escalations works exactly in the same way so we will only take a look at host escalations from now on.

Adding a host escalation

In this guide we will add a small escalation chain that does the following

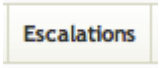
- First notification is sent to the support-group
- After 10 minutes the second (the last one) is sent to the sysadmins group.

To add a host escalation

1 On the start page choose the host you like to add an escalation to in the drop down list.

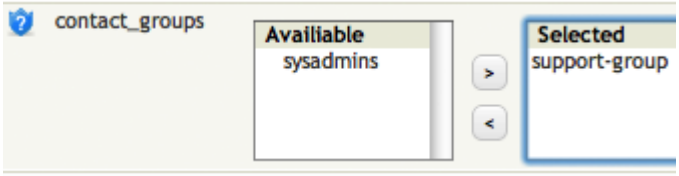
2 Click **Go**.

3 Click Escalations.



4 Add the escalation number one.

a Choose the contact group that shall have the notification.



b Set the start number in the escalation chain.

c Set the end number in the escalation chain [1].

d Set the notification interval which is the number of minutes to wait to the next notification.

e Choose the time period when this escalation will be in use.

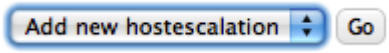
f Choose what states this escalation will be valid for.

In this case we do not use the escalation for unreachable or recovery which means that unreachable and recovery notifications will be sent to the contact group set on the host.

5 Click **Apply changes**.

1.If the start number is 1 and the end number is two it means that the first and the second notification will be handled by this escalation.

6 Choose Add new host escalation



7 Click **Go**.

8 Add the escalation number two.

a Choose the contact group that shall have the notification.

b Set the start number in the escalation chain.

c Set the end number in the escalation chain [1].

d Set the notification interval which is the number of minutes to wait to the next notification. [2]

e Choose the time period when this escalation will be in use.

f Choose what states this escalation will be valid for.

In this case we do not use the escalation for unreachable or recovery which means that unreachable and recovery notifications will be sent to the contact group set on the host.

9 Click **Apply changes**.

10 Click **Save**.

1.We have set the first notification and the last notification to 2 because this escalation will only be used once.
2.The escalation interval is set to 0 because there will be no more escalations when this one is done.

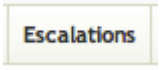
Modifying a host escalation

To modify a host escalation

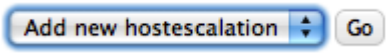
1 On the start page choose the host you like to modify an escalation on in the drop down list.

2 Click **Go**.

3 Click Escalations.



4 Choose the escalation you like to modify.



5 Click **Go**.

6 Make the modifications you like to do and click **Apply changes**.

7 Click **Save**.

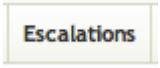
Deleting a host escalation

To delete a host escalation

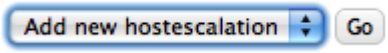
1 On the start page choose the host you like to delete an escalation from in the drop down list.

2 Click **Go**.

3 Click Escalations.

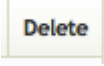


4 Choose the escalation you like to modify.



5 Click **Go**.

6 Click Delete.



7 Click **Save**.

Access rights and contacts

To be able to login to op5 Monitor you need to have a user, described in *Access rights* on page 31. But you need to have a contact, described in *Contacts* on page 29, to be able to receive notifications and in some cases even be able to see any hosts or services.

By connecting access rights to a contact you will be able to login and get notifications with the user created in access rights.

So basically what you need to do is to configure a new contact. Add the contact to an existing contact group or create a new contact group specific for the new contact. If you created a new contact group make sure to add the contact group for the hosts and services that you want to make available in the customized view.

Add new access rights and connect it to the contact you created earlier.

Connecting access rights to contacts

To connect access rights to a contact

1. Configure a new contact.
2. Add the contact to an existing contactgroup or create a new contactgroup specific for the new contact.
   If you created a new contactgroup make sure to add the contact group for the hosts and services that you want to make available in the customized view.
3. Configure a user in access rights with the exact same name as the contact you created.
4. Set the options for the new access right.
   When selecting options do not use the last four options, authorized for all. By doing this the new user will only see the hosts and services that uses the contactgroup that he is a member of.

Make things easy

Profiles

Creating a Profile

To create a profile

1 On the start page choose the host you like to create a profile of in the drop down list.

2 Click Go.

3 Click the Clone button

4 Select the services you wish to include

5 Select Save as Profile

6 Enter name and description for the profile you are creating

7 Click Clone

You are then presented with the option of creating clones based on this new profile. If you do not wish to do this now, you can simply use the left hand web menu to return to Configure or another part of op5 Monitor.

Using a Profile

To use a profile

1 From the start page click Profiles

2 Click use next to the profile

3 Select what parts of the profile you want to include

4 Fill in the number of copies and click Continue...

5 Fill out host details for the clones and click Create

Cloning objects

Cloning from an existing Host

To clone a host

Follow the instructions from in *Creating a Profile* on page 65, except do not click **Save as Profile**.

Cloning services

If you want to create the same service check on multiple host first create the service check on host then clone the service check to one or more hosts.

It is also possible to clone multiple services to one or more hosts or hostgroups.

To clone a service to an other host

1. Choose the Configure web menu.
2. Choose your host you want to copy from, then click 'Go'
3. Click Services for host... in the 'RELATED ITEMS' menu.
4. Select the service (or one of the services) you want to clone then click on Go and then on Clone.
5. Select the service(s) you want to clone.
6. You can chose to clone the service(s) to a list of hosts, a hostgroup or all hosts in a hostgroup.
7. Click Clone.

Propagate settings

To change the same directive on many objects of the same type can be a really time consuming work. This is where the propagate function in op5 Monitor is very handy.

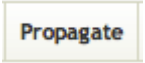
With the propagate function you can copy the value of a directive from one object to one or many other objects of the same type.

In the guide below we will use the propagate function to copy the parents from one host to a couple of other hosts.

To propagate a value of a directive

1. On the start page choose the host you like to propagate a directive value from in the drop down list.

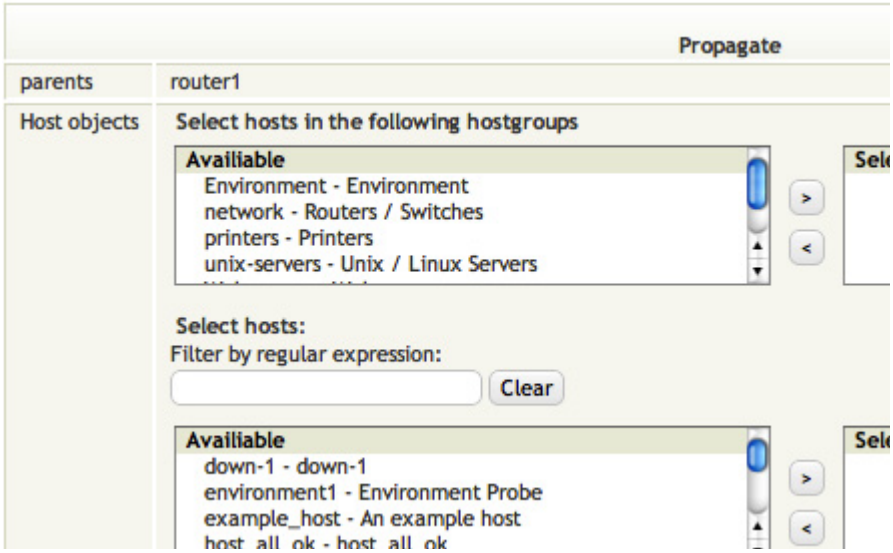
2. Click **Go**.
3. Click **Propagate**.



Now you will see a check box in front of every directive like this.

4. Check the check box for parents and click **Propagate selected settings**.

5 Select the host objects (host or whole host groups) you like to propagate the settings to.



6 Click **Go**.

7 Click **Save**.

Bulk delete

Bulk delete is powerful tool to remove several host or services at once.

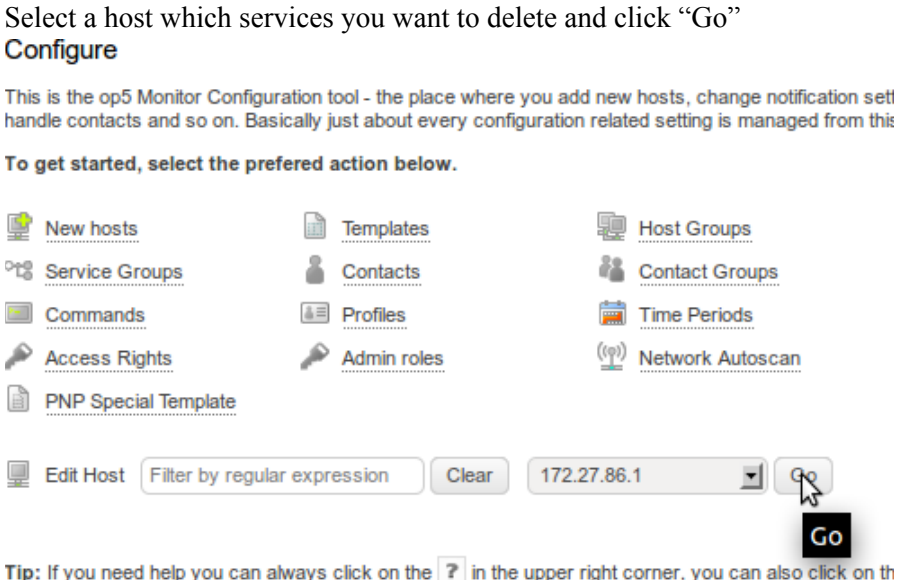
Bulk delete support the following objects:

- Hosts
- Services
- Hostgroups
- Servicegroups
- Contacts
- Contactgroups
- Commands
- Time Periods

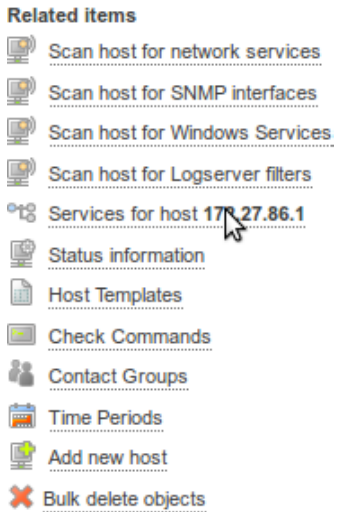
As an example, we will delete two services "Ping" on two different hosts, but the process is similar on all objects listed above.

To delete multiple services this is preformed trough Configure

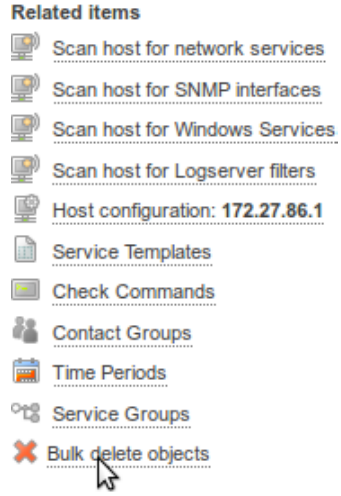
1 Select a host which services you want to delete and click “Go”



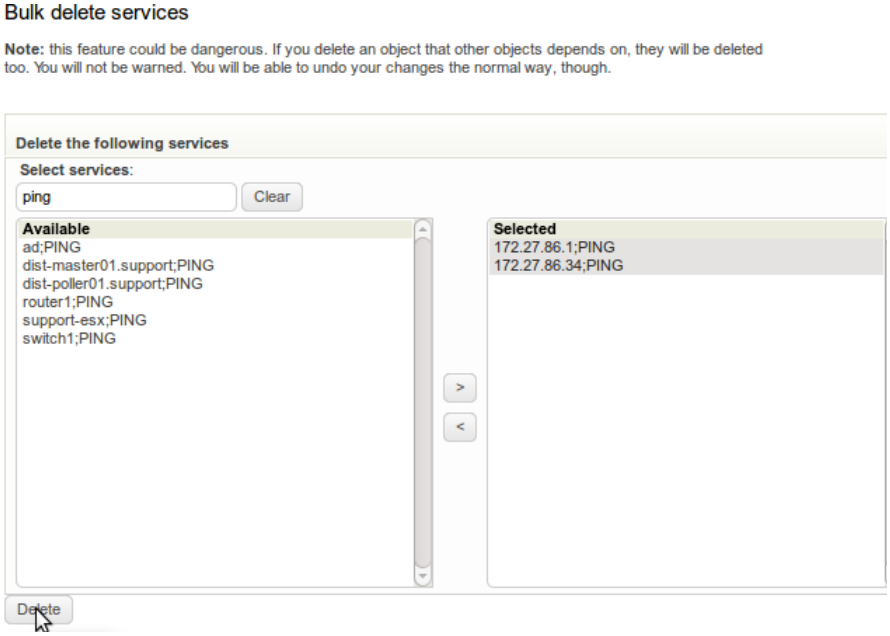
2 Click “Services for host” in the right menu.



3 Click on “Bulk delete objects”



4 Select the services that you want to delete and click “Delete”



5 And click “Save applied changes”

Time periods

Add a time period

Time periods is time defining objects that span over a week. You can define included time for each day of the week in the time period definition.

You can also:

- use already defined time periods as excludes
- add exceptions based on dates and ranges of days

The time period objects are used at many places in the configuration. Most noticeably are in the contact objects where the time periods defines when notifications should be sent out.

You can also use time periods to define when a service or a host should be monitored or when you are creating availability reports.

Macros

Macros can be used to a lot of things. It can for example be used for paths, passwords and retrieving information from op5 monitor.

You can read more about notification commands in *Notification macros* on page 126 in the *Notifiactions* chapter.

Pre-defined macros

By default op5 monitor has a number pre-defined macros. All from path to plugin folder to retrieving information about the last state of service check.

Below is a list of some macros a complete lite of macros can be found at nagios home page: http://nagios.sourceforge.net/docs/3_0/macrolist.html

MACRO	DESCRIPTION
$USER1$	Path to /opt/plugins.
$ARGn$	The nth argument passed to the command
$HOSTNAME$	Short name for the host.
$HOSTADDRESS$	Address of the host.
$HOSTSTATE$	A string indicating the current state of the host ("UP", "DOWN", or "UNREACHABLE").
$SERVICEDISPLAYNAME$	An alternate display name for the service.
$SERVICESTATE$	A string indicating the current state of the service ("OK", "WARNING", "UNKNOWN", or "CRITICAL").

Custom macros

It is possible to create your own macros. This can be used to store passwords or user names for example.

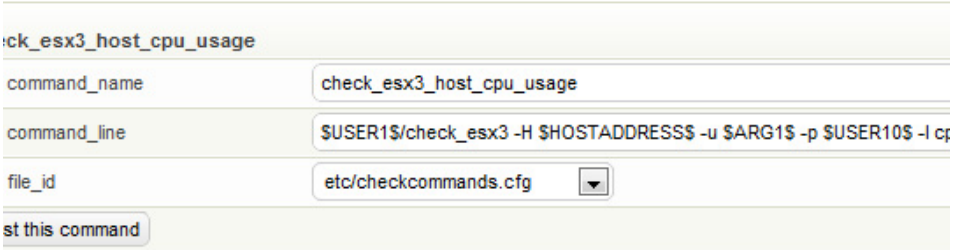
All custom macros should be put in the file /opt/monitor/etc/resource.cfg

A custom macro should use the $USERn$ macro.

To define a password for a check, first add the macro in resource.cfg

```
# Password for vmware user
$USER10$=secretpassw0rd
```

After that add the macro to check command, in this example we use the check_esx3_host_cpu_usage check command.




```
command_name=check_esx3_host_cpu_usage
command_line=$USER1$/check_esx3 -H $HOSTADDRESS$ -u $ARG1$ -p
$USER10$ -l cpu -s usage -w $ARG2 -c $ARG3$
```

This check will use the following macros:

$HOSTADDRESS$ - Will get the address of the host from the configuration

$ARG1$ - Use the fist argument from the check command.

$USER10$ - Use the argument specified in resources.cfg with the same name.

$ARG2$ - Use the second argument from the check command.

$ARG3$ - Use the third argument from the check command.

Features not supported by Configure

Even though some features are not supported by the op5 Monitor configuration tool you can still use them.

The hostgroup_name is one of them.

What you have to do is to add a separate configuration file not read by the import function in Configure. Then you add your other configuration tricks into that file.

To add a configuration file not read by Configure

1. Open up a ssh connection to the op5 Monitor server and login as root.
2. Create the following file with an editor of your choice:
   `/opt/monitor/op5/nacoma/custom_config.php`
3. Add the following code to the file you just created:
   ```
   <?php
   $notouch_file_prefix = "_";
   ?>
   ```
4. Create a configuration file with "_" as a prefix to the file name like this:
   `touch /opt/monitor/etc/_custom_objects.cfg`
5. Add the file to the `/opt/monitor/etc/nagios.cfg` with by adding the following line below the other cfg_file variables in `nagios.cfg`:
   `cfg_file=/opt/monitor/etc/_custom_objects.cfg`
6. Restart op5 Monitor.
   `service monitor restart`

Now you may add your objects to the new configuration file and they will not be loaded into Configure. But you can still see the objects using View config as it is described in the op5 Monitor user manual.

Plugins

About plugins

This chapter covers the following topics:

Introduction

op5 Monitor is shipped with many plugins that cover most monitoring needs. But what to do if one of your corporate applications can not be monitored straight out of the box?

Often you can find a plugin at www.nagiosexchange.org, and since op5 Monitor and Nagios uses the same plugin format you can often simply download a plugin, put it in /opt/plugins/custom/ and start using it.

However, if you can not find a suitable plugin anywhere you might have to write your own plugin. Since the plugin interface is very straight-forward, anyone with a fair amount of UNIX scripting experience can do this.

Paths and macros

All standard plugins shipped with op5 Monitor is installed in:

```
/opt/plugins
```

The macro you use to reach the plugins folder is:

```
$USER1$
```

The plugins you add to the system by your own must be placed in:

```
/opt/plugins/custom
```

And they will then be reached with the following macro/path:

```
$USER1$/custom
```

The reason for placing your own plugins in /opt/plugins/custom is because then it will not be touched by any upgrade from op5.

Before you start

Be for you can start develop you own plugins you need to make sure you have ssh access or terminal access to your op5 server the possibility to transfer files to your op5 Monitor server any kind of editor, vim and jed are installed by default on your op5 Monitor server.

Microsoft windows users may use PuTTY for terminal access via SSH and WinSCP for file transfers via SFTP (SSH).

Macintosh or UNIX/Linux users may use the commands ssh or scp from a local terminal window.

The plugin interface

A plugin is a small executable that takes optional command line parameters as input and

1. Performs a test
2. Reports a diagnostic message on stdout (will be shown in the web gui)
3. Returns an exit code.

Example 1 Execute check_tcp to test the port 80/tcp on 193.201.96.136

```
monitor!root:~# /opt/plugins/check_tcp -H 193.201.96.136 -p 80
TCP OK - 0.043 second response time on port
80|time=0.042824s;0.000000;0.000000;0.000000;10.000000
monitor!root:~# echo $?
0
monitor!root:~# /opt/plugins/check_tcp -H 193.201.96.136 -p 143
Connection refused
monitor!root:~# echo $?
2
monitor!root:~#
```

In the *Example 1* on page 79 we first execute check_tcp to test that port 80/tcp on 193.201.96.136 responds, which it does, hence the exit code of 0.

Then we check port 143/tcp on the same host and that port is not open, hence the result is Critical - exit code 2.

The result output is actually built upon two parts divided by a | sign (pipe). The text on the

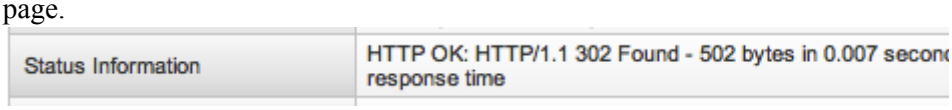
- left hand side of the | is the status information
- right hand side of the | is the performance data.

The performance data is not mandatory but you need it if you want your plugin to be able to produce graphs for you in op5 Monitor.

Status output

The Status output is the text describing the result in readable words. The plugin must print the status output to stdout when your plugin is executed.

You will see it in the Status state information on the Service or Host information page.



This text can be anything, including HTML, you like to use to describe the status situation for your plugin.

Performance data

The performance data is data displaying the result in numbers. The plugin must print the status output to stdout when your plugin is executed. It is also used to produce performance graphs in op5 Monitor.

So if you want graphs from your plugin you need to have performance data in your output.

The performance data is setup like this:

```
'label'=value[UOM];[warn];[crit];[min];[max]
```

Table 1 Performance parts with descriptions.

Part	Description
`label`	The label can contain any characters. If space is included quotes are needed.
`value`	The plugin was able to check the service, but it appeared to be above some "warning" threshold or did not appear to be working properly
`UOM`	Can be any of: • no unit assuming an integer as a value • s - seconds (also us, ms) • % - percentage. • B- Bytes (also KB, MB, GB and TB) • c - A continuous counter like bytes transmitted on an interface.
`warn, crit, min, max`	• Can all be null and trailing unfilled semicolons can be dropped. • min and max is not needed if UOM is %. • value, warn, crit, min and max must be of the same UOM.

Example 2 Performance data output

```
time=0.218901s;;;0.000000 size=42236B;;;0
```

The *Example 2* on page 81 shows a performance data output from a plugin with two values separated with one space in the output.

Return code

The return code is the one that op5 Monitor uses to determine what state the services is in. It may be one of the following:

`0, 1, 2, 3`

All above 0 is to be known as **problem states**.

Table 2 The return codes in detail.

Nr	Name	Description
0	Ok	The check did ok and everything seems to be working fine.
1	Warning	The plugin was able to check the service, but it appeared to be above some "warning" threshold or did not appear to be working properly
2	Critical	The plugin detected that either the service was not running or it was above some "critical" threshold
3	Unknown	Something unknown happened during the check. Things like invalid command line arguments or low-level failures internal to the plugin shall **not** be reported as Unknown state.

Adding your first plugin to op5 Monitor

In this section we will create a very simple plugin. We will write it as a bash script in a ssh connection to the op5 Monitor server.

This plugin will not actually be very useful but we will use it to describe the steps needed when you starts to add other more useful plugins.

Creating the plugin

To create a simple example plugin as a bash script

1
```
cd /opt/plugins/custom
touch helloworld
chmod 755 helloworld
```

2 Open up the script with your favorite text editor and type in the following example plugin:

```
#!/bin/sh
echo 'WARNING: Hello world!'
exit 1
```

3 Save and exit your editor

4 Execute it from the terminal:

5
```
./helloworld
WARNING: Hello world!
echo $?
1
```

The script prints the status output (WARNING: Hello world!).

`echo $?` prints the return code of the last executed command.

Configuring op5 Monitor to use the plugin

To configure op5 Monitor to use the plugin

1 Go to **Configure** and chose **Commands**.

2 Add a new command with:
```
command_name: check_local_helloworld
command_line: $USER1$/custom/helloworld
```

3 Click **Apply** and then **Save**.

Now you may use your check command with a service.

Creating a more complex plugin

In this section we will create a more complex and useful plugin compared to the one we created in *Adding your first plugin to op5 Monitor* on page 83. We will stick to bash, because of the simplicity

We will create a plugin that checks that the storage path specified in /etc/op5backup.conf exists, to make sure that op5backup.sh is configured properly for local operation.

To create a more complex plugin

1 Create the script and editing it:

```
cd /opt/plugins/custom
touch check_op5backup
chmod 755 check_op5backup
```

2 Open up the script with your favorite text editor and type in the following code:

```
#!/bin/bash
# Create a function to print the storage path
storagepath() {
grep ^storagepath /etc/op5backup.conf |
tail -1 |
sed 's/^[^"]*"//g' | sed 's/"$//g'
}

# Put the storage path in an environmental variable
STORAGEPATH=`storagepath`

# Test if the storagepath exists and is a directory
if [[ ! -d "$STORAGEPATH" ]]; then
 # Print a warning message for the web gui
 echo op5backup.sh is not properly configured for local
operation
# Exit with status Warning (exit code 1)
exit 1
fi

# If the script reaches this point then the test passed
# Print an OK message
echo $STORAGEPATH exists
# Exit with status OK
exit 0
```

3 Add a check_command like this using the op5 Monitor web gui:
command_name: `check_op5backup`
command_line: `$USER1/custom/check_op5backup`

4 Enter the service configuration for your monitor server, and add a service with `check_op5backup` as the check_command.

5 Save configuration.

More information

This chapter has only scratched on the surface of how to write your own plugins.

To read more about plugin development take a look at the **Nagios plugin development guidelines**:

http://nagiosplug.sourceforge.net/developer-guidelines.html

Widgets

About widgets

This chapter covers the following topics:

Introduction

In the op5 Monitor user manual we describes how the widgets works in the user interface. There you can read about how to

- hide and show them
- move around the widgets
- change the widgets refresh rate
- restore to default settings.

In this chapter we will take a look at how you could create your own widgets.

The op5 Monitor user interface is using Kohana as framework some of the back end parts for the widgets are handled by Kohana. But this chapter will only describe the widget development in it self.

In this chapter we will focus on creating a small hello world widget.

The widget basics

The widget helper

A widget is instantiated using the widget helper:

```
widget::add('netw_health', array('index', $this->model), $this);
```

The helper takes care of a lot of the back end stuff like handling paths and assigning master controller variables. More on this later.

Why do we use a helper and not a controller?

Since Kohana fires off some system events (like `system.post_controller`) every time a controller is finished, we do not want this overhead. Also, we do not want the widget output to be returned to browser immediately but wanted the master (calling) template to decide where to render the output. Our solution is instead to use a helper class and buffer the output using `ob_start()` and `ob_get_contents()`.

The widget rules

All widgets need to follow a few rules. They need to have a correct

- file structure
- constructor
- index method
- view (template)
- return

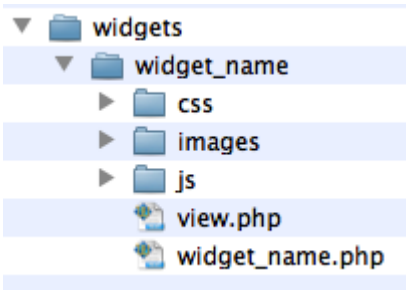
File structure

The widget has to be placed in a folder with the exact name as the widget.

All default widgets shipped with op5 monitor is placed in:

```
/opt/monitor/op5/ninja/application/widgets
```

And the file structure below the widgets folder looks like this



But as we are developing our own widget we need to place it in:

`/opt/monitor/op5/ninja/application/custom_widgets`

The widget class should be named like:

```
widget_name + _Widget
```

and should extend widget_Core. Since our widget will be called hello world the class will look like this:

```
class Hello_world_Widget extends widget_Core {
    //
}
```

Everything is case sensitive here.

Constructor

Our widget also needs a constructor:

```
public function __construct(){
    parent::__construct();
    # needed to figure out path to widget
    $this->set_widget_name(__CLASS__, basename(__FILE__));
}
```

The constructor sets the name and path needed for the master controller to be able to render the widget correctly.

To look at our example again:

```
widget::add('netw_health', array('index', $this->model), $this);
```

The second argument to `widget::add()` is an array and the first element should always be the method to call. Some widgets will need access to some data to be useful and in this case we are passing a reference to the model used in the master controller.

The last argument here is the reference to the master controller so that we are able to pass the generated widget content. The master controller takes care of putting these at the appropriate place in the rendered HTML head.

Index method

Our index method:

```
public function index($arguments=false, $master=false){
    $this->master_obj = $master;
}
```

This first assignment is important since it is how we refer back to the master controller - without it, the widget won't be visible anywhere.

The widgets view file could be called anything (as long as it has a php extension) and to use it you call:

```
$view_path = $this->view_path('view');
```

with the name of the view template as an argument. In this case the template file will be called:

```
view.php
```

View (template)

The entire widget output should be contained in a div

```
<div class="widget movable collapsable removable closeconfirm"
id="widget-helo_world">
  // widget content goes here
</div>
```

You should give the div an ID following the format:

```
'widget_'<widget_name>
```

to make it possible to access it in a generic way using Java Script.

To add extra css resources you assign `$this->css` with an array like this:

```
$this->css = array('/css/hello_world');
```

The same thing is done with needed Java Script files using `$this->as`.

After writing the code that actually does something (left out here as we are only focusing on the widget in it self) there is only couple of more things to do.

Here we are now requiring the actual template file into the widget class file.

```
require_once($view_path);
```

Return data

Last thing needed is return `$this->fetch();` which will do the `ob_get_contents()` thing and pass everything back to the calling controller.

The actual location of the widget on the rendered page, is then all up to the master controller and template - the widget should only return a view.

Writing a simple widget

In this example we will create a small hello_world widget. We will assume you have:

- ssh access to the server
- required knowledge about PHP
- knowledge about how to use an editor in a Linux environment.
- access to the mysql database

We does also assume that you, before you start, log in to the op5 Monitor server.

Creating the directory structure

To create the directory structure

1 Go to the application folder:
```
cd /opt/monitor/op5/ninja/application
```

2 Create the following folders:
```
mkdir -p custom_widgets/hello_world
mkdir custom_widgets/hello_world/css
mkdir custom_widgets/hello_world/images
mkdir custom_widgets/hello_world/js
```

Writing the widget file

To write the widget file

1 Go to the widget folder:
```
cd /opt/monitor/op5/ninja/application/custom_widgets/
hello_world
```

2 Create a file, with your favorite editor, called:
```
hello_world.php
```

3 Type in the following content in the file:
```
<?php defined('SYSPATH') OR die('No direct access allowed.');
class Hello_world_Widget extends widget_Core{
   public function __construct() {
      parent::__construct();

      # needed to figure out path to widget
      $this->set_widget_name(__CLASS__, basename(__FILE__));
   }

   public function index($arguments=false, $master=false) {
      $this->master_obj = $master;

      # fetch widget view path
      $view_path = $this->view_path('view');
```

```
        # Set the title of the widget.
        $title = "Hello World";

        # Give the widget an id.
        $widget_id = $this->widgetname;

        # fetch widget content
         require_once($view_path);

        return $this->fetch();
    }
}
?>
```

4 Save and exit from your editor.

Writing the view file

To write the view file (template file)

1 Go to the widget folder:
```
cd /opt/monitor/op5/ninja/application/custom_widgets/
hello_world
```

2 Create a file, with your favorite editor, called:
```
view.php
```

3 Type in the following content in the file:
```
<?php defined('SYSPATH') OR die('No direct access allowed.');
?>
<div class="widget movable collapsable removable closeconfirm"
id="widget-<?php echo $widget_id ?>">
     <div class="widget-header"><?php echo $title ?></div>
     <div class="widget-content">
          <!-- START: widget content here  -->
               Hello World
          <!-- END: widget content -->
     </div>
</div>
```

4 Save and exit from your editor.

Adding the widget to the widget table

Be fore we can see the widget on the tactical overview we need to add it to the widgets table in mysql.

To add the widget to the database

1 Open up the mysql console, as the root user.

2 Execute the following query:

```
INSERT INTO `ninja_widgets` (`user`, `page`, `name`,
`friendly_name`, `setting`) VALUES ('', 'tac/index',
'hello_world', 'Hello World', '');
```

3 Exit from the mysql console.

Viewing the widget

If everything is done correctly we will now be able to view are first simple op5 Monitor widget.

To view the widget

Open up the Tactical overview in the op5 Monitor user interface and it will look like this:

Hello World

Refreshing in the background

All widgets shipped with op5 Monitor supports editing if enabled (by setting class '`editable`') in the wrapping div. By using Java Script it is possible to get background (Ajax) refresh with just a few lines of code.

In the Hello widget widget (hello_world) a java script resource is added in the hello_world.php file. It will point to a java script file located in the `js` folder in the widget folder.

To add background refresh

1. Go to the widget folder:
   ```
   cd /opt/monitor/op5/ninja/application/custom_widgets/
   hello_world/js
   ```
2. Create a file, with your favorite editor, called:
   ```
   hello_world.js
   ```
3. Type in the following content:
   ```
   $(document).ready(function() {
       var netw_health = new widget('hello_world', 'widget-
   content');
   });
   ```
4. Save and close the editor.
5. Go one step up in the folder hierarchy and open up hello_world.php
6. Add the following line in the index method somewhere between line 14 and 24:
   ```
   $this->js = array('/js/hello_world');
   ```

As you can see, it's only one line of code wrapped in the document ready event listener. It creates a new instance of the widget Java Script class with 2 arguments:

- the widget identifier (widget name)
- what CSS class in the widget that should be updated by the Ajax call.

By adding this code, a JQuery UI slider will be created to be used when editing update frequency. The element with the CSS class `widget-content` in the example above will be updated with the interval set by the slider (default 60 sec).

Take your widget a step further

The example we have been working on here in this chapter is very basic and the output is not much to use. Once you have understood the basics you will probably like to create a more useful widget.

One way to get more information about how you can create a more advanced widget is to take a look at one of the widgets shipped with op5 Monitor.

The Network health widget is a good example. That one can be found here:

```
/opt/monitor/op5/ninja/application/widgets/netw_health
```

If you are changing any of the default widgets remember to create a copy of the widget, with a new nam, and place it in:

```
/opt/monitor/op5/ninja/application/custom_widgets/
```

Packaging your widget

To make it easy for other users to install and start using your widget you should make a package of it. Then one can install the package in the Tactical Overview in the op5 Monitor GUI.

The package is actually a normal zip file that contains

- the widget in it self
- manifest.xml

The manifest.xml file contains basic data needed by op5 Monitor so that it knows how to install the widget.

Creating the Manifest.xml

To create a Manifest.xml file

1. Create an xml file that looks like this:
```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Manifest file for widget to be used in Ninja4Nagios  -->
<widget_content>
<author>John Doe</author>
<version>1.0</version>
<Friendly_name>My cool widget</Friendly_name>
<description>A cool widget for op5 Monitor Tactical
Overeview.</description>
<page>tac/index</page>
</widget_content>
```
2. Place it in the folder where your widget is located and make sure it is called: manifest.xml

Creating the widget package

Now your widget should be ready and located in the filesystem of your op5 Monitor server.

In this instruction we assume that your widget is called:

```
my_own_widget
```

To create a widget package

1. Go to the folder above the my_own_widget folder where your widget is located.
2. Create the package with the zip command like this
```
zip -r my_own_widget.zip my_own_widget/
```

Access widgets externally

To access a widget from an external site, like an intranet or network status page.

Server side setup

To configure this you need to configure a user, edit php-settings for external widgets and insert an iframe on your external web site.

Contact configuration

To set up a widget contact you first need to create a access right. When that is configured create a contact with the same name as the access right and specify which contact groups(s) is should be a member of.

Same as a normal contact the contact groups defines which hosts is visible.

PHP-settings

After the contact is created log in to the op5 Monitor server using SSH.

```
# /opt/monitor/op5/ninja/application/config/external_widget.php /
opt/monitor/op5/ninja/application/config/custom
```

```
# cd /opt/monitor/op5/ninja/application/config/custom
```

Edit the file external_widget.php with your favorite editor.

This file has two variables, “widget_name” specifies which widget that should be shown by default if no widget is set in the iframe. The next one is “username” and this sets the user that should be allowed to fetch the widget.

When we set a user name here that user will no longer be able to login to op5 Monitor and will only be a “widget user”.

Example:

```
$config['widget_name'] = 'netw_health';
```

```
$config['username'] = 'jsmith';
```

This example the contact jsmith will be used to view widgets and by default it will show the network health widget.

External website setup

On the external website you will need to add an iframe in which the widget is displayed.

The format of the iframe look like this:

```
<iframe src="http://<SERVER_NAME>/ninja/index.php/
external_widget/show_widget/<OPTIONAL WIDGET_NAME>" height="500px"
frameborder=0 width="600px" scrolling='no'></iframe>
```

In this iframe you will need to change the <SERVER_NAME> to you op5 monitor host name and <OPTIONAL WIDGET_NAME> can either be removed and the default widget will be used or you can specify a widget name to view another widget.

The widgets names can we found in the folder /opt/monitor/op5/ninja/application/widgets. The folder names is the same as the widget name.

GUI themes

About GUI themes

This chapter covers the following topics:

Subject	Page	Subsections
Introduction	102	
The files and folders	103	
Make your own theme	105	*Before you start* on page 105 *Creating your own theme* on page 105 *Changing what theme op5 Monitor use* on page 105
Making changes in the user interface	106	*Changing the logo* on page 106 *Adding hostname to the Quick bar* on page 106 *Change the default font* on page 107

Introduction

All views in the op5 Monitor user interface are built up with help of theme templates. In the default op5 Monitor installation there is only one them:

```
default
```

If you like to change any of the parts of the op5 Monitor user interface the best way to do that is to create your own theme. That makes sure you do not lose any changes in an upgrade later on. Of course you have to update your theme by your self to be able to enjoy many of the new features that comes with op5 Monitor updates.

In this chapter we will take a closer look of how the theme is built up and some minor changes that can easily be made.

The files and folders

An op5 Monitor theme includes a lot of folders and files. Most of them are never a subject to be changed but you still need them.

All themes shall be placed in a folder of its own directly under:

`/opt/monitor/op5/ninja/application/views/themes`

All views have their own folder named after the controller they belong to. The view folders can contain everything from one single PHP file to a complex structure of folders, code files (PHP, java script, css), images etc.

Almost all controllers have their corresponding view in the user interface.

Beside the view folders we have the following folders and files:

- `admin/`
- `css/`
- `css_header.php`
- `error.php`
- `icons/`
- `js/`
- `js_header.php`
- `kohana_unit_test.php`
- `login.php`
- `menu.php`
- `ninja_start.php`
- `template.php`
- `unauthorized.php`

A more detailed description of the files listed above is shown in the table below.

File/Folder	Description
admin/	For future functions in the user interface. Not in use at the moment.
css/	CSS files that is used by the controllers. The controller it self decides what file to use.
css_header.php	Locates and enables the files in `css/` for the controllers.  Do not touch this file!
error.php	A general error messages template.
icons/	All icons used in the user interface.
js/	Java scripts that is used by the controllers. The controller it self decides what file to use.

File/Folder	Description
js_header.php	Locates and enables the files in `js/` for the controllers.  Do not touch this file!
kohana_unit_test.php	 Do not touch this file!
login.php	The user interface login page.
menu.php	Deprecated!
ninja_start.php	Deprecated!
template.php	The template file in it self. This is the one that creates the main parts of the user interface.
unauthorized.php	A general unauthorized messages template.

Make your own theme

Before you start

Before you can start making changes to the please make sure you have

- ssh and sftp access to the op5 Monitor server
- created your own theme.

In all instructions in the rest of the chapter we assume you already have logged in via ssh on the op5 Monitor server. We also assume that you have the basic knowledge needed in PHP and knows how to work in a Linux environment.

The theme we create here will be called:

```
my_theme
```

Creating your own theme

To create your own theme

1. Go to the theme folder:
   ```
   cd /opt/monitor/op5/ninja/application/views/themes
   ```
2. Copy the default theme to a new directory with the name of your new theme:
   ```
   cp -a default/ my_theme
   ```

Changing what theme op5 Monitor use

To change what theme op5 Monitor shall use

1. Go to the application folder:
   ```
   cd /opt/monitor/op5/ninja/application/config
   ```
2. Open up `config.php` in your favorite text editor.
3. Look up and change the following line and change the theme name from `default/` to `my_theme/` in this case:
   ```
   $config['current_theme'] = 'default/';
   ```
4. Save and exit.

Making changes in the user interface

As you probably already have realized you can do almost any kind of changes in the op5 Monitor user interface. Covering them all would require a complete manual of its own. So in this chapter we will only take a look at a few of them.

- Changing the logo.
- Adding hostname to the Quick bar.
- Change the default font.

The topics listed above should give you knowledge to do other modifications by your own.

Changing the logo

One thing you might want to do is to change the default logo up in the left corner of the user interface.

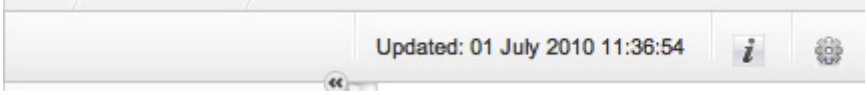
Before you starting

To change the logo

1. Make sure you have followed the instructions in:
   Make your own theme on page 105.
2. Create your own logo file. It shall meet the following criteria:
   Width: 19px
   Height: 19px
   Type: PNG
3. Replace the following file with your own:
   `/opt/monitor/op5/ninja/application/views/themes/my_theme/icons/icon.png`

Adding hostname to the Quick bar

If you have more then one op5 Monitor server it might be a bit difficult to remember which one you are logged in to when your working in the user interface. Then it could be a good idea to add the hostname to the Quick bar at the top of the user interface.



To add the hostname to the Quick bar

1. Make sure you have followed the instructions in:
   Make your own theme on page 105.

2 Go to the folder of your theme:
```
cd /opt/monitor/op5/ninja/application/views/themes/my_theme
```

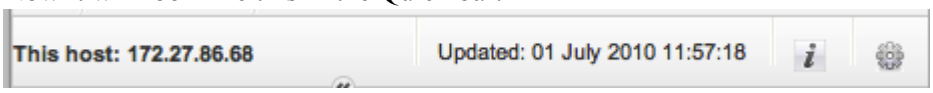
3 Open up `template.php` in your favorite text editor.

4 Look up the following lines (starting on row 196 if you have not changed the default file):
```
<div id="quicklinks">
</div>
```

5 Between the lines found in *4* add the following line:
```
<br /><strong>This host: <?php echo $_SERVER['SERVER_NAME'];
?></strong>
```

6 Save and exit from the editor.

Now it will look like this in the Quick bar:



Change the default font

Many visual parts in op5 Monitor are setup in css files. There fore its a good idea to take a look at them and see how they are used.

In this example we will change the default font and make it a bit bigger.

To change the default font

1 Make sure you have followed the instructions in:
Make your own theme on page 105.

2 Go to the folder of your theme:
```
cd /opt/monitor/op5/ninja/application/views/themes/my_theme
```

3 Open up the `css/default/common.css` file in your favorite text editor.

4 At the top of the file common.css you will find the following lines:
```
* {
text-decoration: none;
font-size: 1.0em;
outline: none;
padding: 0;
margin: 0;
}
```

Change `font-size: 1.0em;` to:
```
font-size: 1.02em;
```

5 Save and exit from the editor.

6 Refresh the op5 Monitor user interface in your browser and you can see that the default font is a bit bigger now.

User menus

About user menus

This chapter covers the following topics:

Subject	Page	Subsections
Custimize user menus	52	

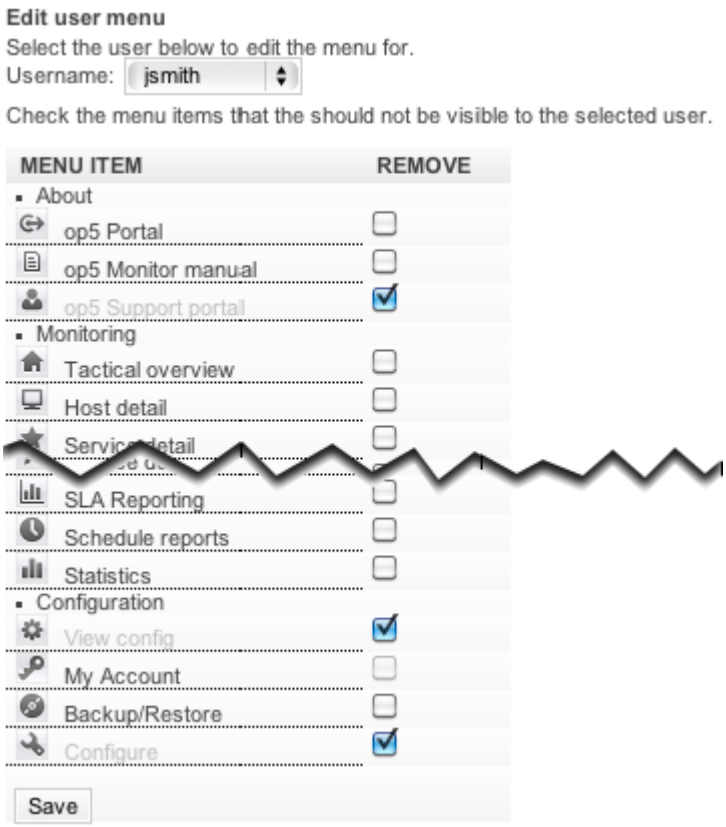
Custimize user menus

It is possible for a administrator to custimize users menu.

Only user with full access can edit user menus.

To change a specific users menu, go to ‘My Account’ in the menu and click on ‘Edit user menu’. Select the user you want to change the menu for.

You can now hide the options in the menu that you don’t want to be visible for that specific user. In the example below we have removed ‘op5 Support portarl’, ‘View Config’ and ‘Configure’ options.



When you are done, click on save.

Localization

About Localization

This chapter covers the following topics:

Subject	Page	Subsections
Introduction	110	
Downloading and starting the tools	111	
Adding a new language	112	*Changing basic language file settings* on page 113 *Applying the new language to the server* on page 114

Introduction

In op5 Monitor we have the possibility to show all texts in your own language. This so called localization is done with help from gettext, which is a part of the Linux translation project.

There are many different ways to work with the gettext files. You can use

- the cli commands from the gettext installation
- a program like poedit.

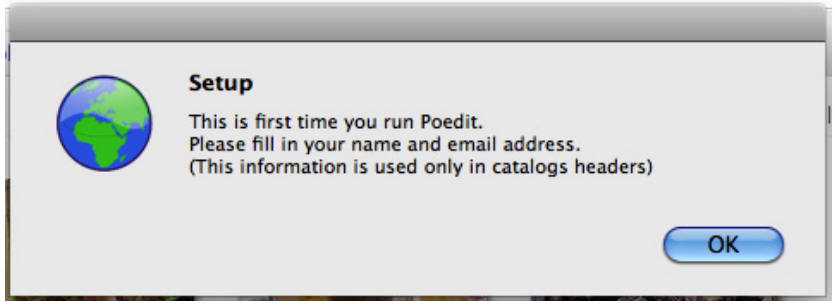
Here in this chapter we will use poedit to add a new language to op5 Monitor.

Downloading and starting the tools

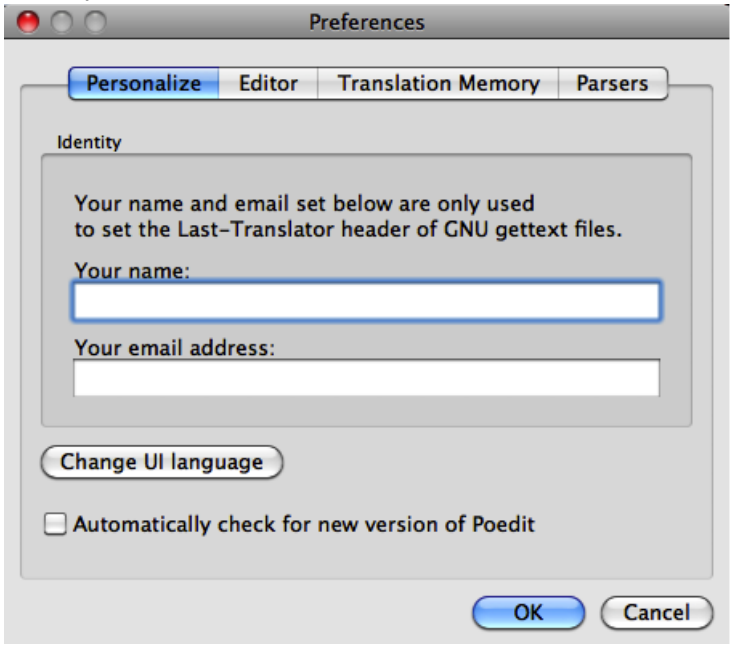
The first thing we need to do is to download the tools. In this case just only the Poedit

To download and starting the tools:

1. Go to:
   http://www.poedit.net/download.php
2. Download and install the version needed for you OS.
3. Start poEdit and follow the instructions.
   a. Click **OK**



   b. Fill in your name and email address and click **OK**.

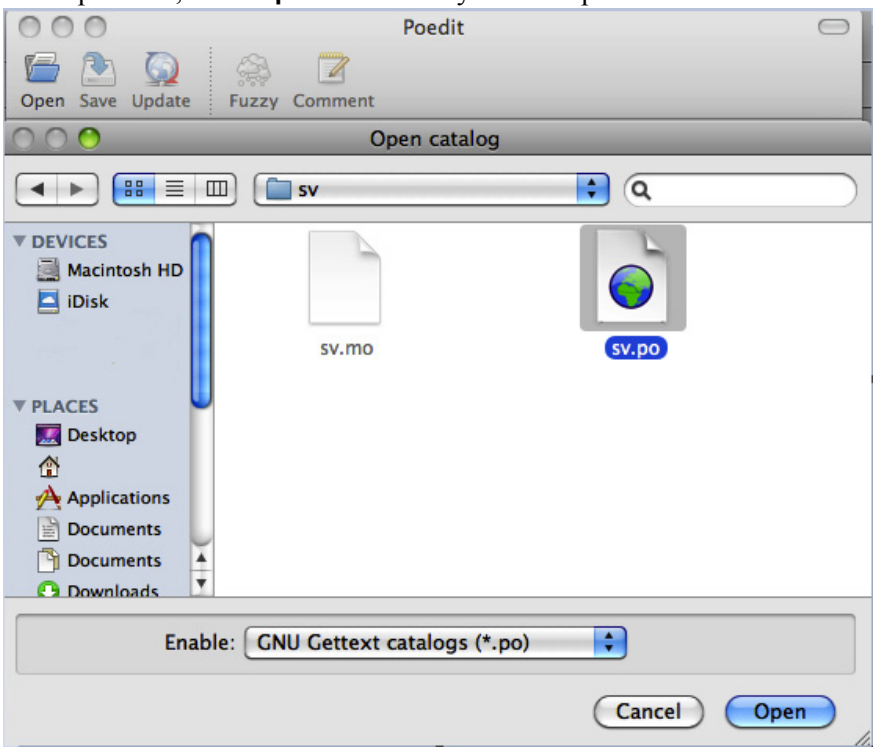


Adding a new language

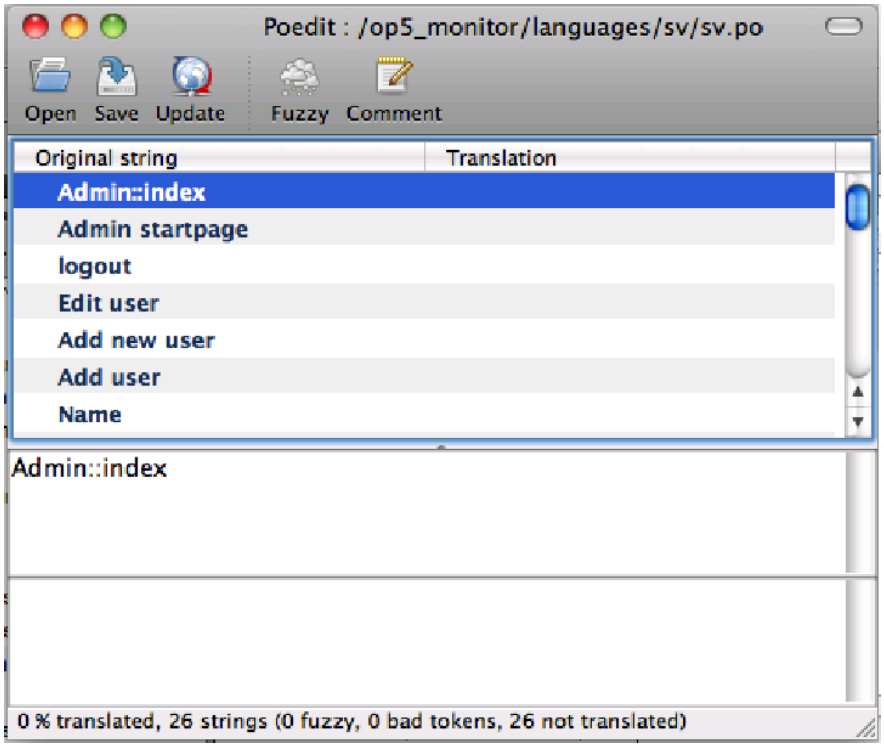
In this example we will create add a new language (swedish). We will also use the Poedit tool and work with the files locally on our workstation and then copy the new language files to the op5 Monitor server.

To add a new language

1. Copy the language folder to your workstation.
2. Create a new folder inside the language folder with the short name of the language you like to create.
   ```
   mkdir sv/
   ```
3. Copy the files from the en/ folder (english) to the new one and rename them to the same name as the folder like this:
   ```
   cp en/en.mo sv/sv.mo
   cp en/en.po sv/sv.po
   ```
4. Start up Poedit, click **Open** and locate your new .po file.



5 Mark the line you like to add your translation to and type in the field at the bottom of the window.



6 Save the file when done.

Changing basic language file settings

Now the last thing we shall do, before we upload the files to the server, is to change some of the basic settings in the language file.

The things we are going to change is:

- Team
- Language
- Country

To change the basic language file settings

1 Open up the language file (sv.po, in this example) in Poedit.

2 In the main menu click **Catalog** -> **Settings...**

3 Change the **Team** to what ever you want and the **Language** and **Country** to reflect the language you are creating. In this example:
Language: `Swedish`
Country: `SWEDEN`

4 Click **OK**.

Applying the new language to the server

Now the last thing we need to do is to send up the new language to the server.

To apply the new language to the server

1. Copy the new language folder (in this example `sv/`) and its content to the following folder on the op5 Monitor server:
   `/opt/monitor/op5/ninja/application/languages/`
2. Open up your browser and change the settings so the new language will be the first one to use.
   In FireFox this is done in:
   Preferences -> **Content** -> **Languages**
3. Go to the op5 Monitor user interface login page. If you have translated all lines you will now see the login page in your new language.

Graph templates

About graph templates

This chapter covers the following topics:

Subject	Page	Subsections
Introduction	116	
PNP web front end	117	
Pages	118	
Templates	120	*What are templates?* on page 120 *What template will be used when?* on page 120 *Creating own templates* on page 121

Introduction

op5 Monitor is using PNP to create the graphs available for most standard services in the user interface.

PNP is an add-on to nagios which analyzes performance data provided by plugins and stores them automatically into RRD-databases (Round Robin Databases).

PNP only processes performance data built according to the Developer Guidelines for nagios plugins. With this limitation we want to honour the work of Nagios Plugin Developers who stick to the guidelines.

This is a short description of how to use PNP and it's functions pages and templates.

For more info please refer to the online manual for pnp

http://www.pnp4nagios.org/pnp/start

Kudos to **Joerg Linge** for letting us use his text.

PNP web front end

The behavior of the PNP Web-Front end can be controlled through the config file `/opt/monitor/etc/pnp/config.php`.

This file will be overwritten during updates of PNP as the paths and options are detected during `./configure`.

Own adjustments should be made in:

`/opt/monitor/etc/pnp/config_local.php`

If this file does not exist the file `config.php` can be taken as a guideline.

To access the PNP web front end through the GUI click on ‘Graphs’ in the menu.

Pages

Pages provide the opportunity to collect graphs of different hosts and services on one page. That way - as an example - you can display the traffic rates of all tape libraries.

Regular expressions are possible so you can accomplish a lot with only few definitions - provided that you have appropriate names.

The directory specified using `$conf['page_dir']` contains one or more file with the extension

`.cfg`.

The file name (without the extension) appears in the list of available pages and will be used as title of the browser window.

Comments start with a hash-sign (`#`) and are possible within lines as well.

Each file contains a page definition which specifies the name of the page and it determines whether the following graph definition contains regular expressions or not.

```
define  page  {
        suse_regex 1# 0 = use no regexp, 1 = use regexp
        page_name test-page# page description
}
```

One or more `graph` definitions follow:

```
define graph {
        host_name       host1,host2,host3
        service_desc    Current_Load
}
define graph {
        host_name       host4
        service_desc    Current_Users
}
```

And now some definitions with regular expressions. At first all hosts whose names are starting with `Tape`:

```
define graph {
        host_name       ^Tape
        service_desc    Traffic
}
```

all hosts whose names are ending with `00`:

```
define graph {
        host_name       00$
        service_desc    Load
}
```

all services of localhost whose names contain `a` or `o`, respectively:

```
define graph {
        host_name       localhost
        service_desc    a|o
}
```

all services whose names contain an underscore followed by (at least) three digits on all hosts whose names start with `UX`:

```
define graph {
        host_name       ^UX
        service_desc    _\d{3}
}
```

Templates

What are templates?

PNP uses templates to influence the appearance of RRD graphs. The selected check command determines which template will be used to control the graph.

Following will be described where templates are stored and how the decision for the “right” template is made.

What template will be used when?

Templates are stored at two places in the file system.

- /opt/monitor/op5/pnp/templates.dist, for templates included in the PNP package
- /opt/monitor/op5/pnp/templates, for custom made template which are not changed during updates

If the graph for the service `http` on host `localhost` should be shown, PNP will look for the XML file

```
perfdata/localhost/http.xml
```

and read its contents. The XML files are created automatically and contain information about the particular host and service.

The header contains information about the plugin and the performance data. The XML tag

```
<TEMPLATE>
```

identifies which PNP template will be used for this graph.

/localhost/http.xml

```
<NAGIOS>
  <DATASOURCE>
    <TEMPLATE>check_http</TEMPLATE>
    <DS>1</DS>
    <NAME>time</NAME>
    <UNIT>s</UNIT>
    <ACT>0.006721</ACT>
    <WARN>1.000000</WARN>
    <CRIT>2.000000</CRIT>
    <MIN>0.000000</MIN>
    <MAX></MAX>
  </DATASOURCE>
  <DATASOURCE>
    <TEMPLATE>check_http</TEMPLATE>
    <DS>2</DS>
```

```
    <NAME>size</NAME>
    <UNIT>B</UNIT>
    <ACT>263</ACT>
    <WARN></WARN>
    <CRIT></CRIT>
    <MIN>0</MIN>
    <MAX></MAX>
  </DATASOURCE>
...
</NAGIOS>
```

PNP will look for a template with the name check `http.php` in the following sequence:

1. `templates/check_http.php`
2. `templates.dist/check_http.php`
3. `templates/default.php`
4. `templates.dist/default.php`

The template `default.php` takes an exceptional position as it is used every time no other applicable template is found.

Creating own templates

PNP templates are PHP files which are included during execution of PNP using the PHP function `include()`. This means that every PHP code in templates will be interpreted so manipulation of all values is possible.

PNP template must have the following characteristics:

1. templates must contain valid PHP code.
2. templates must not create any output.
3. the two arrays `$opt[]` and `$def[]` have to be filled.

These two arrays are used to call **rrdtool graph** so every option is possible that RRDtool supports. All options of RRDtool are described very thoroughly on the RRDtool Home page:

http://oss.oetiker.ch/rrdtool/doc/rrdgraph.en.html

If both arrays contain more than one set of data graphs will be created for every set.

Inside the templates the data from the related XML files can be used.

Using the relatively simple template response.php we will describe the most important options.

```
<?php
#
$opt[1] = "--title \"Response Time For $hostname / $servicedesc\"
";
```

```
#
$def[1] =  "DEF:var1=$rrdfile:$DS[1]:AVERAGE " ;
$def[1] .= "AREA:var1#00FF00:\"Response Times \" " ;
$def[1] .= "LINE1:var1#000000 " ;
$def[1] .= "GPRINT:var1:LAST:\"%3.4lg %s$UNIT[1] LAST \" ";
$def[1] .= "GPRINT:var1:MAX:\"%3.4lg %s$UNIT[1] MAX \" ";
$def[1] .= "GPRINT:var1:AVERAGE:\"%3.4lg %s$UNIT[1] AVERAGE \" ";
?>
```

Note: As the number (`1`) and the letter `L` look alike in this listing: the format `%3.4lg` contains a small letter.

- `$opt[1] = ”-title …`
  sets RRDtool options for the first set of data, here the title as you can see. Embedded quotes are masked using a backslash (`\`).
  The variables `$hostname` and `$servicedesc` were determined through the call of PNP and are available for the template as well.
- `$def[1] = “DEF:var1=$rrdfile:$DS[1]:AVERAGE ”;`
  defines which data is to be read from which RRD file.
  `$rrdfile` contains the path to the RRD file of this service.
  `$DS[1]` refers to the first data series from the RRD file.
- `$def[1] .= “AREA:var1#00FF00:\”Response Times \” ”;`
  the operator `.=` appends more data to the array `$def[1]`.
  An area will be drawn using data from the variable `var1`.
  The color is defined in HEX notation `#00FF00` (red, green, blue).
  The label is `Response Times`.
- `$def[1] .= “LINE1:var1#000000 ”;`
  As completion of the just drawn area a line (`LINE1`) will be drawn in black (`#000000`).
- `$def[1] .= “GPRINT:var1:LAST:\”%3.4lg %s$UNIT[1] LAST \” ”;`
  `$def[1] .= “GPRINT:var1:MAX:\”%3.4lg %s$UNIT[1] MAX \” ”;`
  `$def[1] .= “GPRINT:var1:AVERAGE:\”%3.4lg %s$UNIT[1] AVERAGE \” ”;`
  The three `GPRINT` lines build up the caption for the graph. The current values are formatted using the **printf** syntax.

Notifiactions

About notifications

This chapter covers the following topics:

Subject	Page	Subsections
Introduction	124	
How does notifications works?	125	*Notification filters* on page 125 *Notification commands* on page 125 *Notification macros* on page 126
Notification skins	128	*The content of a notification skin* on page 128 *Creating custom notification skins* on page 129
Dial up notification	131	*Adding a dial up notification command* on page 131 *Configuring the contacts* on page 132
SNMP trap notifications	133	*Adding SNMP notification commands* on page 133 *Configuring the contacts* on page 134

Introduction

In this chapter we will take a deeper look at the notification function in op5 Monitor. We will look at how the

- notification works
- notification skins works (mail/sms/htmlpost)
- dial up notification works
- snmp trap notification works.

How does notifications works?

In the op5 Monitor user manual we describe some of the basics with notifications. Let us take a closer look at how it really works.

Notification filters

When a notification is about to be sent it has to go through a number of filters before op5 Monitor can determine whether a notification really is suppose to be sent or not.

Table 1 Notification filters

Filter	Description
Program-wide	This tells op5 Monitor if notifications are turned on or not in a program-wide basis.
Service and host filters	• Is the host or service in scheduled downtime or not? • Is the host or service in a flapping state? • Does the host or service notification options says that this type of notification is supposed to be sent? • Are we in the right time period for notifications at the moment? • Have we already sent a notification about this alert? Has the host or service remained in the same non-OK state that it was when the last notification went out?
Contact filters	• Does the contacts notifications options says that this type of notification is supposed to be sent? • Are we in the right time period for notifications at the moment, according to the notification time period set on the contact?

Notification commands

How the notifications are sent is defined in either one of the two files below:

- checkcomands.cfg
- misccommands.cfg

The commands are divided into

- host notification commands

- service notification commands

The notification commands are then using scripts in the same way as the normal check commands does.

All default scripts shipped with op5 Monitor is located in:

`/opt/monitor/op5/notify`

Notification macros

Many of the arguments sent to the notification commands are macros. The macros are a sort of variables containing a, in most cases, program-wide value. You can read more about macros in the Nagios manual:

http://nagios.sourceforge.net/docs/3_0/macros.html

One of the most important macro used with notifications is:

`$NOTIFICATIONTYPE$`

This macro tells you what type of notification that is supposed to be sent. The `$NOTIFICATIONTYPE$` macro can have one of the following values.

Table 2 Notification types

Notification type	Description
PROBLEM	A service or host has just entered (or is still in) a problem state.
RECOVERY	A service or host has recovered from a problem state.
ACKNOWLEDGEMENT	A service or host in a problem state has been acknowledged by a user.
FLAPPINGSTART	The host or service has entered a flapping state.
FLAPPINGSTOP	The host or service has left a flapping state.
FLAPPINGDISABLED	The host or service flapping detection has stopped and has there fore left the flapping state.
DOWNTIMESTART	The host or service has entered a scheduled downtime.
DOWNTIMESTOP	The host or service has left a scheduled downtime.
DOWNTIMECANCELLED	The scheduled downtime for a host or service has been cancelled.

The list of macros described in the Nagios manual is very useful when you are working with new notification commands and scripts. That list can be found here:

http://nagios.sourceforge.net/docs/3_0/macrolist.html

Notification e-mail sender

Notifications are by default sent from the e-mail address ”op5monitor” without any domain. The MTA adds the local domain name, witch by default is “@localhost.localdomain”.

To change the e-mail address that notification are sent from use the --from-mail argument for the notification command.

To change the sender e-mail address from op5monitor@localhost.localdomain to op5notification@mycompany.com simply go to the check command for the host-notify and add “--from-email op5notification@mycompany.com” without the “-signs.

```
command_name=host-notify
command_line=$USER3$/notify/poller_notify_send.pl --from-email
op5notification@mycompany.com -c "$CONTACTNAME$" -h "$HOSTNAME$" -
f "$NOTIFICATIONTYPE$" -m "$CONTACTEMAIL$" -p "$CONTACTPAGER$"
"HOSTALIAS=$HOSTALIAS$" "HOSTADDRESS=$HOSTADDRESS$"
"HOSTSTATE=$HOSTSTATE$" "HOSTSTATEID=$HOSTSTATEID$"
"HOSTSTATETYPE=$HOSTSTATETYPE$" "HOSTATTEMPT=$HOSTATTEMPT$"
"HOSTLATENCY=$HOSTLATENCY$"
"HOSTEXECUTIONTIME=$HOSTEXECUTIONTIME$"
"HOSTDURATION=$HOSTDURATION$" "HOSTDURATIONSEC=$HOSTDURATIONSEC$"
"HOSTDOWNTIME=$HOSTDOWNTIME$"
"HOSTPERCENTCHANGE=$HOSTPERCENTCHANGE$"
"HOSTGROUPNAME=$HOSTGROUPNAME$" "HOSTGROUPALIAS=$HOSTGROUPALIAS$"
"LASTHOSTCHECK=$LASTHOSTCHECK$"
"LASTHOSTSTATECHANGE=$LASTHOSTSTATECHANGE$"
"LASTHOSTUP=$LASTHOSTUP$" "LASTHOSTDOWN=$LASTHOSTDOWN$"
"LASTHOSTUNREACHABLE=$LASTHOSTUNREACHABLE$"
"HOSTOUTPUT=$HOSTOUTPUT$" "HOSTPERFDATA=$HOSTPERFDATA$"
"HOSTACKAUTHOR=$HOSTACKAUTHOR$" "HOSTACKCOMMENT=$HOSTACKCOMMENT$"
"NOTIFICATIONNUMBER=$NOTIFICATIONNUMBER$"
"CONTACTALIAS=$CONTACTALIAS$" "DATETIME=$DATETIME$"
"SHORTDATETIME=$SHORTDATETIME$" "DATE=$DATE$" "TIME=$TIME$"
"TIMET=$TIMET$" "HOSTACTIONURL=$HOSTACTIONURL$"
"HOSTNOTESURL=$HOSTNOTESURL$" "ADMINPAGER=$ADMINPAGER$"
"ADMINEMAIL=$ADMINEMAIL$"
"NOTIFICATIONCOMMENT=$NOTIFICATIONCOMMENT$"
```

This has to be done for the command “service-notify” as well.

Notification skins

The three basic notifications (email, sms and htmlpost notifications) are all using something called notification skins. The notification skins are templates describing how the notification is supposed to look like when it is sent to its receiver.

If we will take a look at the notify folder we will find the following skins folders:

- `skins.htmlpost/`
- `skins.mail/`
- `skins.sms/`

Each folder contains a number of notification skins divided into host and service notification filters.

- `host.ACKNOWLEDGEMENT`
- `host.FLAPPINGSTART`
- `host.FLAPPINGSTOP`
- `host.PROBLEM`
- `host.RECOVERY`

- `service.ACKNOWLEDGEMENT`
- `service.FLAPPINGSTART`
- `service.FLAPPINGSTOP`
- `service.PROBLEM`
- `service.RECOVERY`

As you can see there is one skin for the most common notification types.

The content of a notification skin

Let us take a look at what a skin looks like.

Example 1 *The sms service.PROBLEM skin*

```
#SERVICEDESC# on #HOSTNAME# is #SERVICESTATE#. #SERVICEOUTPUT#
```

This is a very simple skin. The reason for that is that you can not send too much data with a normal sms.

Example 2 *The mail service.PROBLEM skin*

```
From: op5Monitor
To: #CONTACTEMAIL#
Subject: [op5] #NOTIFICATIONTYPE#: '#SERVICEDESC#' on '#HOSTNAME#'
is #SERVICESTATE#

#extra_host_vars#

op5 Monitor
```

```
Service #NOTIFICATIONTYPE# detected #LASTSERVICESTATECHANGE#.
'#SERVICEDESC#' on host '#HOSTNAME#' has passed the #SERVICESTATE#
threshold.

#STATUS_URL#

Additional info;

#SERVICEOUTPUT#

Host:    #HOSTNAME#
Address: #HOSTADDRESS#
Alias:   #HOSTALIAS#
Status:  #HOSTSTATE#
Comment: #NOTIFICATIONCOMMENT#

Service: #SERVICEDESC#
Status : #SERVICESTATE#
Latency: Check was #SERVICELATENCY# seconds behind schedule
Misc   : Check took #SERVICEEXECUTIONTIME# seconds to complete

Additional links (requires configuration);

Host actions: #HOSTACTIONURL#
Host notes:   #HOSTNOTESURL#Service actions: #SERVICEACTIONURL#
Service notes:   #SERVICENOTESURL#
```

The mail notifications can contain a lot more data and there we add a lot more to the mail skin file.

In both *Example 1* on page 128 and *Example 2* on page 128 you find text like:

- `#SERVICEDESC#`
- `#HOSTNAME#`

That text is called **keywords**.

The keywords will be replaced with the value of a command line argument looking like this:

```
FOO=BAR
```

So a command line argument like the one above will generate a keyword with the name `FOO` having the value `BAR`.

Note: If a notification macro, or other value sent to a corresponding keyword, is missing in the notification command it will not stop the notification from being sent. It is only the replacement that will be missing.

Creating custom notification skins

Sometimes the default notification skins needs to be changed. This shall not be done in the default folders.

To create custom notification skins

1. Go to the notify folder:
   `cd /opt/monitor/op5/notify`
2. Create the custom-skins folder:
   `mkdir custom-skins`

3 Copy the skins.* folders to the custom-skins folder:
```
cp skins.* custom-skins/
```

4 Make the changes you like to do and the new skins will be used at directly after you have saved the changes.

Dial up notification

Many of the modern mobile phones are only giving you one tiny signal when a sms arrives. If you are on duty during the night you might not wake up or if you are in a very noisy environment it might take some time for you to notice the arrived sms. There for we have included a dial up notification in op5 Monitor.

This is a very simple, but effective, notification that works like this:

Table 3 Dial up notification workflow

Step	Action
1	op5 Monitor is scheduling a notification.
2	The notification goes through all the filters.
3	The `notify_dial.pl` script is called with the following command line: `/opt/monitor/op5/notify/notif_dial.pl <mobilephonenumber>`
4	`notify_dial.pl` is shutting down `smsd`
5	`notify_dial.pl` tries to call the `<mobilephonenumber>` If the line is busy or no one answer the call in 45 seconds `notify_dial.pl` will hang up and try again two more times before it quits.
6	The user answer the call and `notify_dial.pl` hangs up.
6	`notify_dial.pl` is starting up `smsd` again and the execution is over.

Adding a dial up notification command

This is done in two steps:

- add the command
- configure the contacts

To add a dial up notification command

1. Login to the op5 Monitor user interface and go to **Configure**.
2. Click **Commands**.
3. Add a new command with the following settings:
   command_name `notify_by_dial`
   command_line `$USER3$/notify/notify_dial.pl "$CONTACTPAGER$"`
4. Click **Apply**.
5. Click **Save**.

Configuring the contacts

To configure the contacts

1. Login to the op5 Monitor user interface and go to **Configure**.
2. Either open up an existing contact and create a new one.
3. On the contact set **Pager** to a phone number on the form like this (***without*** the leading '+'-sign):
   `46705123123`
4. Set **host_notification_commands** and **service_notification_commands** to:
   `notify_by_dial`
5. Click **Apply**.
6. Click **Save**.

Note: Make sure the contact is a member of the contact_group is associated with the correct objects.

SNMP trap notifications

op5 Monitor is shipped with the possibility to send notifications as SNMP traps. To start use the SNMP notifications you need to

- add a few new commands
- configure the contacts

Adding SNMP notification commands

Here we need to add two commands one for host notifications and one for service notifications.

To add a SNMP notification command

1. Login to the op5 Monitor user interface and go to **Configure**.
2. Click **Commands**.
3. Add the following new commands with the following settings:
   command_name `host_notify_by_snmp`
   command_line `$USER3$/notify/notify_by_snmp.pl -H` *`snmp.trap.host`* `-C` *`SNMPCOMMUNITY`* `-t nHostNotify`
   `"NOTIFICATIONTYPE=$NOTIFICATIONTYPE$"`
   `"NOTIFICATIONNUMBER=$NOTIFICATIONNUMBER$"`
   `"HOSTACKAUTHOR=$HOSTACKAUTHOR$"`
   `"HOSTACKCOMMENT=$HOSTACKCOMMENT$" "HOSTNAME=$HOSTNAME$"`
   `"HOSTSTATEID=$HOSTSTATEID$" "HOSTSTATETYPE=$HOSTSTATETYPE$"`
   `"HOSTATTEMPT=$HOSTATTEMPT$"`
   `"HOSTDURATIONSEC=$HOSTDURATIONSEC$"`
   `"HOSTGROUPNAME=$HOSTGROUPNAME$"`
   `"LASTHOSTCHECK=$LASTHOSTCHECK$"`
   `"LASTHOSTSTATECHANGE=$LASTHOSTSTATECHANGE$"`
   `"HOSTOUTPUT=$HOSTOUTPUT$"`

   command_name `service_notify_by_snmp`
   command_line `$USER3$/notify/notify_by_snmp.pl -H` *`snmp.trap.host`* `-C` *`SNMPCOMMUNITY`* `-t nSvcNotify`
   `"NOTIFICATIONTYPE=$NOTIFICATIONTYPE$"`
   `"NOTIFICATIONNUMBER=$NOTIFICATIONNUMBER$"`
   `"SERVICEACKAUTHOR=$SERVICEACKAUTHOR$"`
   `"SERVICEACKCOMMENT=$SERVICEACKCOMMENT$" "HOSTNAME=$HOSTNAME$"`
   `"HOSTSTATEID=$HOSTSTATEID$"`
   `"SERVICEDESCRIPTION=$SERVICEDESCRIPTION$"`
   `"SERVICESTATEID=$SERVICESTATEID$"`
   `"SERVICEATTEMPT=$SERVICEATTEMPT$"`

```
"SERVICEDURATIONSEC=$SERVICEDURATIONSEC$"
"SERVICEGROUPNAME=$SERVICEGROUPNAME$"
"LASTSERVICECHECK=$LASTSERVICECHECK$"
"LASTSERVICESTATECHANGE=$LASTSERVICESTATECHANGE$"
"SERVICEOUTPUT=$SERVICEOUTPUT$"
```

Change the following to their correct value, in both commands:
snmp.trap.host
SNMPCOMMUNITY

4 Click **Apply**.

5 Click **Save**.

Configuring the contacts

To configure the contacts

1 Login to the op5 Monitor user interface and go to **Configure**.

2 Either open up an existing contact och create a new one.

3 Set **host_notification_commands** to:
`host_notify_by_snmp`

4 Set **service_notification_commands** to:
`service_notify_by_snmp`

5 Click **Apply**.

6 Click **Save**.

Note: Make sure the contact is a member of the contact_group is associated with the correct objects.

LDAP authorization

About LDAP authorization

This chapter covers the following topics:

Introduction

Microsoft Active Directory is used to share user list, provide single sign on and other central features in large Microsoft based workstation and server networks.

Active Directory is Microsoft's implementation of existing business standards such as LDAP, Kerberos and DNS.

The purpose of this chapter is to provide a step by step guide on how to integrate op5 Monitor authentication with Microsoft Active Directory.

Before we start

To be able to complete this how-to you will need:

- Administrator access to a working Microsoft Active Directory environment.
- Root command line access to a running op5 Monitor.

Do not use space in the admin group name

Please refrain from using spaces in the admin group name, as this can cause problems.

If your admin group and users reside in an OU containing spaces in its name, you will need to manually edit two files after accepting the new configuration.

Preparing your Active Directory

Before configuring op5 Monitor, we need to set up a user op5 Monitor can use to read authentication data from Active Directory, and an Admin group for the op5 Monitor itself.

To prepare your Active Directory

1. Create a normal user:
   `op5viewer`
2. Create a global group for admin rights to op5 Monitor:
   `op5admin`.
3. Add an AD-user to the `op5admins` group.

Let us say that we have an AD structure that looks like this

```
webinar.op5
|
|-- op5_Operators (OU)
|   |
|   |-- admins (Group)
|   |-- op5admin (User, member of admins)
|   |-- viewers (Group)
|   `--op5viewer (User, member of viewers)
|
|
`-- Users (Default AD Container)
    |
    `-- kalle.kula (User, member of admins)
```

Now make sure you have your configuration files like this:

```
/opt/op5sys/etc/ldapserver:
LDAP_BASE=dc=webinar,dc=op5 LDAP_BIND_DN=op5auth@webinar.op5
LDAP_GROUP=ou=op5_Operators,dc=webinar,dc=op5
LDAP_IS_AD=1
LDAP_SERVER=172.27.86.97
LDAP_UPNSUFFIX=webinar.op5
LDAP_USERKEY=uid
LDAP_USERS=ou=op5_Operators,dc=webinar,dc=op5

/opt/op5sys/etc/ldaprights.cfg:
authorized_for_system_information admins,viewers
authorized_for_configuration_information admins,viewers
authorized_for_system_commands admins
authorized_for_all_services admins,viewers
authorized_for_all_hosts admins,viewers
authorized_for_all_service_commands admins
authorized_for_all_host_commands admins
```

```
/etc/httpd/conf.d/op5ldapauth.conf:
<Location /monitor.old/>
AuthzLDAPServer 172.27.86.97
AuthzLDAPUserBase ou=op5_Operators,dc=webinar,dc=op5
AuthzLDAPGroupBase ou=op5_Operators,dc=webinar,dc=op5
AuthzLDAPUserKey sAMAccountName
AuthzLDAPBindDN op5auth@webinar.op5
AuthzLDAPBindPassword p4ssw0rd.
require valid-user
AuthzLDAPUserScope subtree
AuthType basic
AuthzLDAPMethod ldap
AuthName "op5 Monitor Access"
</Location>
```

Now, the "LDAP_USERS" variable in /opt/op5sys/etc/ldapserver controls where update-users.php will find users, and the "LDAP_GROUP" controls where the script will find the groups you have defined in /opt/op5sys/etc/ldaprights.cfg.

So, if I have:

```
LDAP_GROUP=ou=op5_Operators,dc=webinar,dc=op5
LDAP_USERS=ou=op5_Operators,dc=webinar,dc=op5
```

The script will only sync over users that exist under the op5_Operators OU, belonging to any group defined /opt/op5sys/etc/ldaprights.cfg under that OU, so using the above example structure, the cgi.cfg will be populated like this:

```
authorized_for_system_information=op5admin,op5viewer
authorized_for_configuration_information=op5admin,op5viewer
authorized_for_system_commands=op5admin
authorized_for_all_services=op5admin,op5viewer
authorized_for_all_hosts=op5admin,op5viewer
authorized_for_all_service_commands=op5admin
authorized_for_all_host_commands=op5admin
```

See that despite the user kalle.kula being a member of admins, he does not get included, because the script doesn't look for users in the Users Container, only in the op5_Operators OU.

Now, say that we want to have all our users in the User Container, and just keep two groups in the op5_Operators OU, we can change the `LDAP_USERS` variable in `/opt/op5sys/etc/ldaprights.cfg` to:

```
LDAP_USERS=cn=Users,dc=webinar,dc=op5
```

With the above configuration, this results in the following `cgi.cfg`:

```
authorized_for_system_information=kalle.kula
authorized_for_configuration_information=kalle.kula
authorized_for_system_commands=kalle.kula
authorized_for_all_services=kalle.kula
authorized_for_all_hosts=kalle.kula
authorized_for_all_service_commands=kalle.kula
authorized_for_all_host_commands=kalle.kula
```

As you see, now the users from the op5_Operators OU don't get included, despite being members of the correct groups, because we told the script to look for users in the Users container instead.

Configuring op5 Monitor

For the next steps, you will need root access to the machine running op5 Monitor, either via console, or ssh.

In this example the DC server has the following ip address:

`192.168.1.97`

The DC is running a pretty much out-of-the-box Active Directory structure with:

- domain name: `op5.com`
- all users reside in the OU `Users`.

To configure op5 Monitor

1. Log on as root on op5 Monitor and start the configuration script.
   `op5-authconfig`
2. Now a series of questions will be asked, answer yes to the first question about converting from the old op5 auth system, then chose ad as authentication method.
   Below are the rest of the questions and answers provided for our lab environment.

Question	Answer
LDAP Server	`192.168.1.97`
LDAP Search base	`dc=op5,dc=com`
Where are your user DN:s?	`cn=Users,dc=op5,dc=com`
In what sub tree are your groups located?	`cn=Users,dc=op5,dc=com`
Group for admin access	`op5admins`
Username for the server to connect to AD with	`op5auth@op5.com`
Enter bind password op5auth's password	`*********`

3. Finally, accept the change to your authentication config.

4 Make sure the configuration file looks ok.

```
cat /etc/httpd/conf.d/op5ldapauth.conf

<Location />
AuthzLDAPServer 192.168.1.97
AuthzLDAPUserBase cn=Users,dc=op5,dc=com
AuthzLDAPGroupBase cn=Users,dc=op5,dc=com
AuthzLDAPUserKey sAMAccountName
AuthzLDAPBindDN op5auth@op5.com
AuthzLDAPBindPassword l4bp4SSw0rD
require valid-user
AuthzLDAPUserScope subtree
AuthType basic
AuthzLDAPMethod ldap
AuthName "OP5 Monitor Access"
</Location>
```

5 Now try to login to the op5 Monitor user interface with the user you added to the op5admin group in *To prepare your Active Directory* on page 138.

Local authentication fallback

If for some reason the LDAP server is no longer available we can set up a local authentication fallback.

The different authentication methods is:

Name	Description
Ninja	Local authentication with Ninja (op5 Monitor)
LDAP	LDAP authentication with LDAP/AD
Apache	This will trust the Apache authentication setup. This can be used to authenticate with Kerberos

To set up this you first need to copy the authentication configuration file in witch we add the wanted authentication method with it's fallback method.

```
# cp /opt/monitor/op5/ninja/application/config/auth.php /opt/
monitor/op5/ninja/application/config/custom
```

Edit the file auth.php with your favorite editor.

Edit the "$config['driver']" string to use an array in which we specify the wanted authentication methods.

The format of the string look like this when only using two authentication method:

```
$config['driver'] = array('<authentication method name>' =>
'<Display name>', '<fallback authentication method name>' =>
'<Display name>');
```

The firtst authentication method in the list will be the default method.

For example when setting up a LDAP login with a local fallback the string can look like this:

```
$config['driver'] = array('LDAP' => 'AD', 'Ninja' => 'Local');
```

As we only want to change the authentication method we need to remove everything that we do not want to change. This is important as otherwise new updates might not be implemented.

Also keep the firtst row "<?php defined('SYSPATH') OR die('No direct access allowed.');", this is used for security reasons.

Backup

About backup

This chapter covers the following topics:

Subject	Page	Subsections
Introduction	146	*Backup/Restore actions* on page 146
Backing up the configuration	147	
Restoring a configuration backup	148	

Introduction

The op5 Monitor GUI has got a built-in backup feature. This is not supposed to be a replacement to op5-backup.

The configuration backup is only backing up the op5 Monitor configuration, nothing else.

Backup/Restore actions

In the list of backups the second column is called **ACTIONS**. This is the functions you will find there, from the left to the right:

- View what files are included in the backup.
- Restor the backup
- Delete the backup.

Backing up the configuration

To backup your op5 Monitor configuration

1 Click Backup/Restore in the main menu.

2 Click **Save your current op5 Monitor configuration**.

3 Now your backup is created and can be restored at any time you like.

4 Click the backup archive name to download and save the backup archive somewhere else.

Restoring a configuration backup

To restor a op5 Monitor configuration backup

1 Click Backup/Restore in the main menu.

2 Click restor icon on the configuration backup you like to restore.

Now the backup has been restored.

Upgrade

About upgrade

This chapter covers the following topics:

Introduction

op5 Monitor is upgraded in the same way as the other op5 products. If you have an op5 Appliance system you can read about the upgrade procedure in the op5 Appliance system manual.

This chapter will only cover how to upgrade an op5 Monitor software version.

We will learn how to upgrade with the

- Linux command `yum`
- tar.gz files you may download from our support site.

If you are upgrading from one main version to an other (eg. from version 4 to 5) you need to use the tar.gz files found at our support site.

When upgrading over more than one main version (eg. from version 3 to 5) you shall follow the Upgrade guide found at our support site:

http://www.op5.com/support/downloads/upgrade-guide

Upgrading with yum

To upgrade with yum

1. Login to the op5 Monitor server via ssh as the root user.
2. Check what packages that is pending for upgrade by execute:
   yum check-update
3. If you want to apply the upgraded packages execute:
   yum update

Upgrading with tar.gz files

Before you start with the upgrade you need to make sure you have the login to the download sections at www.op5.com. Otherwise you will not be able to download the tar.gz files.

To upgrade with tar.gz files

1. Download the tar.gz file from www.op5.com/get-op5-monitor/download/download-archive/.
   Find the tar.gz file you need. You might need to open up the Archived files at the bottom of the page.
2. Upload the tar.gz file to the op5 Monitor server.
3. Login to the op5 Monitor server via ssh as the root user.
4. Untar the tar.gz file in the root/ folder.
5. Go to the folder that was extracted from the tar.gz file.
6. Now start the upgrade by executing the following script:
   ./install.sh

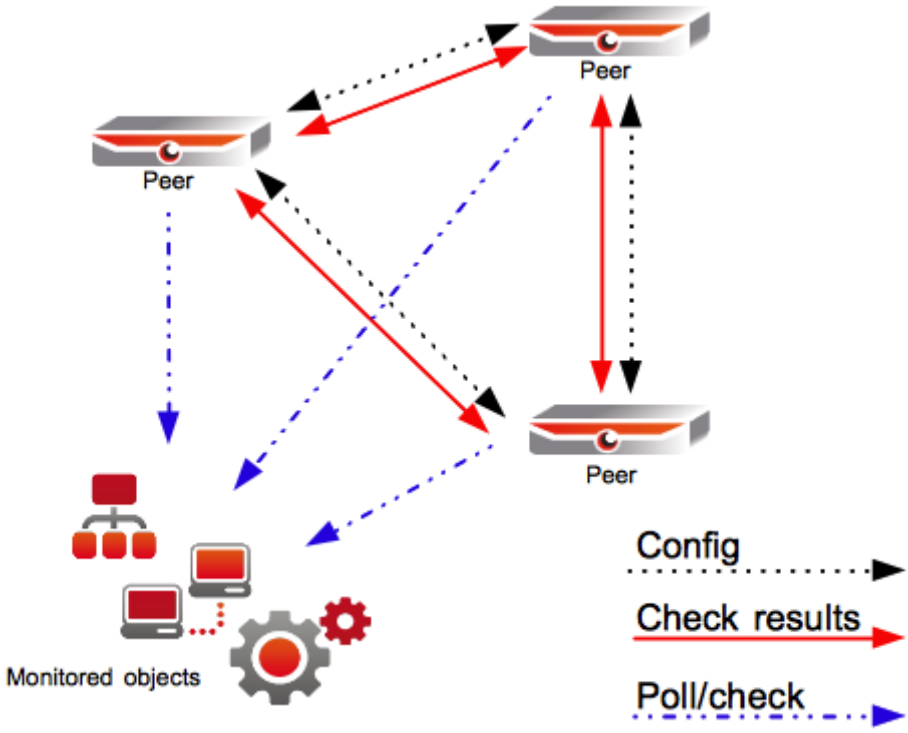
Load balanced monitoring

About the load balanced monitoring

This chapter covers the following topics:

Introduction

The op5 Monitor back end can easily be used as a load balanced monitoring solution. The load balanced model looks like this.



The load balanced solution

- have two or more peers sharing the same task (the hosts to monitor)
- allows configuration at any of the peers
- make sure that all new config is distributed to the peers
- uses the peers to dived the load automatically
- keep tracks of when one peer go down, the other(s) take over the job.

Before we start

There are a few things you need to take care of before you can start setting up an load balanced monitoring. You need to make sure

- you have at least two op5 Monitor servers of **the same architecture** up and running.
- op5 Monitor >=5.2 is installed and running on both machines.
- opened up the following TCP ports for communication between the servers
  - `15551`, op5 Monitor back end communication port
  - `22`, ssh (for the configuration sync).
  - both included servers are to be found in dns or the host file (/etc/hosts).

The configuration

Setting up the load balanced solution

This load balanced configuration will have two so called peers:

- `peer01`
- `peer02`

During the setup we will use the command:

```
mon
```

The mon command is used to make life a bit easier when it comes to setting up a load balanced solution. To get more detailed information about the command mon just execute like this:

```
mon --help
```

To setup a load balanced monitoring solution

1. Log in to one of the systems over ssh, as root.
2. Add the second peer to the configuration with the following command:
   ```
   mon node add peer02 type=peer
   ```
3. Create and add ssh keys to and from the second peer by
   as root user:
   ```
   mon sshkey push --all
   mon sshkey fetch --all
   ```
4. Add peer01 as a peer at peer02
   ```
   mon node ctrl peer02 -- mon node add peer01 type=peer
   ```
5. Make the first initial configuration sync
   ```
   mon oconf push
   ```
6. Restart and push the logs from peer01 to peer02:
   ```
   mon restart; sleep 3; mon oconf push
   ```

Adding a new peer

In this instruction we will have the following hosts:

- `peer01`
- `peer02`
- `peer03` (This is the new one.)

To add a new peer

1. Login to the peer01 as root user over ssh.
2. Add the new peer to the configuration on peer01
   ```
   mon node add peer03 type=peer
   ```

3 Get all ssh keys in place
```
mon sshkey push --all
mon sshkey fetch --all
```

4 Add the peers to one and each other
```
mon node ctrl peer02 -- mon node add peer03 type=peer
mon node ctrl peer03 -- mon node add peer02 type=peer
mon node ctrl peer03 -- mon node add peer01 type=peer
```

5 Manually push the op5 Monitor objects configuration to the new peer.
```
mon oconf push
```

6 Restart monitor on peer01 and send the configuration to all peers again.
```
mon restart ; sleep 3 ; mon oconf push
```

Removing a peer

In this instruction we will remove a peer called:

```
peer02
```

The peer will be removed from all other peers configurations.

To remove a peer

1 Log in to peer01 as root over ssh.

2 Remove all peer configuration from peer02
```
mon node ctrl peer02 -- mon node remove peer01
mon node ctrl peer02 -- mon node remove peer03
```

3 Restart monitor on peer02
```
mon node ctrl peer02 -- mon restart
```

4 Remove peer02 from the rest of the peers, in this case peer03
mon node ctrl --type=peer -- mon node remove peer02

5 Restart the rest of the peers, in this case only peer03
mon node ctrl --type=peer -- mon restart

6 Remove peer02 from the host you are working from.
```
mon node remove peer02
```

7 Restart monitor on the host you are working from.
```
mon node ctrl -- mon restart
```

File synchronization

To synchronize files between servers add a sync paragraph in the file /opt/monitor/op5/merlin/merlin.conf

In the example below we will synchronize the htpasswd.users file to the peer “peer01”

```
peer peer01 {
   address = <ip>
```

```
   port = <port>

   sync {
      /opt/monitor/etc/htpasswd.users /opt/monitor/etc/
htpasswd.users
   }
}
```

Note that this is done per peer.

More information

For more information and a more complex example please take a look at the howto in the git repository of the opensource project of Merlin:

http://git.op5.org/git/?p=nagios/merlin.git;a=blob;f=HOWTO;hb=master#l171

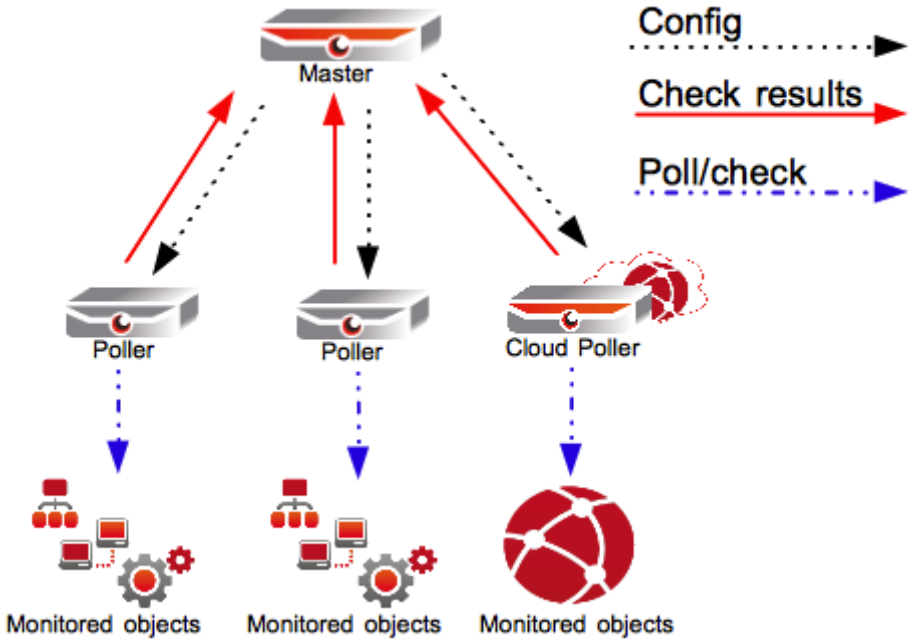
Distributed monitoring

About the distributed monitoring

This chapter covers the following topics:

Introduction

The op5 Monitor back end can easily be configured to be used as a distributed monitoring solution. The distributed model looks like this.



In the distributed monitoring solution

- all configuration is done at the Master
- all new configuration is distributed to the pollers
- each poller is responsible for its own host group (Site).
- the Master has all the status information

Before we start

There are a few things you need to take care of before you can start setting up a distributed monitoring solution. You need to make sure

- you have at least two op5 Monitor servers of **the same architecture** up and running.
- op5 Monitor >=5.2 is installed and running on both machines.
- opened up the following TCP ports for communication between the servers
  - `15551`, op5 Monitor back end communication port
  - `22`, ssh (for the configuration sync).
  - both included servers are to be found in dns.
- Make sure the host group, the one the poller will be responsible for, is added to the master configuration and that at least one host is added to that host group.

The configuration

Setting up the new distributed monitoring solution

This distributed configuration will have one master and one poller:

- `master01`
- `poller01`

The poller will be monitoring the host group gbg.

During the setup we will use the command:

```
mon
```

The mon command is used to make life a bit easier when it comes to setting up a load balanced solution. To get more detailed information about the command mon just execute like this:

```
mon --help
```

To setup a distributed monitoring solution with one poller

1. Log in to the master over ssh, as root.
2. Add the new poller to the configuration with the following command:
   ```
   mon node add poller01 type=poller hostgroup=gbg
   ```
3. Create and add ssh keys to and from the second peer by as root user:
   ```
   mon sshkey push --all
   mon sshkey fetch --all
   ```
4. Add master01 as master at poller01:
   ```
   mon node ctrl --type=poller -- mon node add master01 type=master
   ```
5. Set up the configuration sync:
   ```
   dir=/opt/monitor/etc/oconf
   conf=/opt/monitor/etc/nagios.cfg
   mon node ctrl -- sed -i /^cfg_file=/d $conf
   mon node ctrl -- sed -i /^log_file=/acfg_dir=$dir $conf
   mon node ctrl -- mkdir -m 775 $dir
   mon node ctrl -- chown monitor:apache $dir
   ```
6. To make sure you have an empty configuration on poller01:
   ```
   mon node ctrl -- mon oconf hash
   ```

   This will give you an hash looking like this (“`da 39`” -hash):
   ```
   da39a3ee5e6b4b0d3255bfef95601890afd80709
   ```
7. Now push the configuration to the poller:
   ```
   mon oconf push
   ```
8. Restart and push the logs from master01 to poller01:
   ```
   mon restart; sleep 3; mon log push
   ```

Adding a new poller

In this instruction we will add a new poller to our distributed solution. Here we have the following hosts:

- `master01`
- `poller01`

To add a new poller

1. Log in to the master over ssh, as root.
2. Add the new poller to the configuration with the following command:
   ```
   mon node add poller02 type=poller hostgroup=gbg
   ```
3. Create and add ssh keys for the root user:
   ```
   mon sshkey push poller02
   mon sshkey fetch poller02
   ```
4. Add master01 as master at poller02:
   ```
   mon node ctrl poller02 -- mon node add master01 type=master
   ```
5. Set up the configuration sync:
   ```
   dir=/opt/monitor/etc/oconf
   conf=/opt/monitor/etc/nagios.cfg
   mon node ctrl poller02 -- sed -i /^cfg_file=/d $conf
   mon node ctrl poller02 -- sed -i /^log_file=/acfg_dir=$dir
   $conf
   mon node ctrl poller02 -- mkdir -m 775 $dir
   mon node ctrl poller02 -- chown monitor:apache $dir
   ```
6. To make sure you have an empty configuration on poller01:
   ```
   mon node ctrl poller02 -- mon oconf hash
   ```

   This will give you an hash looking like this ("`da 39`" -hash):
   ```
   da39a3ee5e6b4b0d3255bfef95601890afd80709
   ```
7. Now push the configuration to the poller:
   ```
   mon oconf push
   ```
8. Restart and push the logs from master01 from poller01:
   ```
   mon restart; sleep 3; mon oconf push
   ```

Adding a new host group to a poller

You might want to add an other host group for to a poller. You need to edit the merlin.conf file to do that. This is not doable with any comand as it is today.

To add new host group to a poller

1. Open up and edit `/opt/monitor/op5/merlin/merlin.conf`.
2. Add a new host group in the hostgroup line like this:
   ```
   hostgroup = gbg,sth,citrix_servers
   ```

   Remember to not put any space between the hostgroup name and comma.

3 Restart monitor on the poller
```
mon restart
```

4 Send over the new configuration to the poller
```
mon oconf push
```

Removing a poller

In this instruction we will remove a poller called:

```
poller01
```

The poller will be removed from the master configuration and all distributed configuration on the poller will also be removed.

To remove a poller

1 Log in to the master over ssh, as root.

2 Deactivate and remove all distributed setup on the poller host.
```
mon node ctrl poller01 -- mon node remove master01
```

3 Restart monitor on the poller.
```
mon node ctrl poller02 -- mon restart
```

4 Remove the poller from the master configuration.
```
mon node remove poller01
```

5 Restart monitor on the master.
```
mon restart
```

Master takeover

If a poller goes down the default configuration is for the master to take over all the checks from the poller. For this to work all hosts monitored from the poller most also be monitorable from the master.

If the master server not should take over the checks from the poller this can be set in the merlin configuration file.

To stop the master from taking over, edit the file /opt/monitor/op5/merlin/merlin.conf

Add the following to the poller that you want the master not to take over.

```
takeover = no
```

Note that this is done per poller.

File synchronization

To synchronize files from the master server to the poller add a sync paragraph in the file /opt/monitor/op5/merlin/merlin.conf

In the example below we will synchronize the htpasswd.users file to the poller “poller01”

```
poller poller01 {
   address = <ip>
   port = <port>
   contact_group = <contactgroup>

   sync {
      /opt/monitor/etc/htpasswd.users /opt/monitor/etc/
htpasswd.users
   }
}
```

Note that this is done per poller

One way connections

If one peer is behind some kind of firewall or is on a NAT adress it might not be possilbe for the master server to connect to the peer.

To tell the master not to connect to the poller and let the poller open the session we need to add a option to the file /opt/monitor/op5/merlin/merlin.conf.

Under the section for the poller that the master should not try to connect to add the following:

```
connect = no
```

Example

In the example below we have a master “master01” that can not connect to “poller01” but “poller01” is allowed to connect to “master01”.

```
poller poller01 {
   address = <ip>
   port = <port>
   contact_group = <contactgroup>
   connect = no
   }
```

Is is also possible to set this option on the peer instead then the master will always initiate the session.

Recovery

After a poller as been unavailable for a master (i.e of network outage) the report data will be synced from the poller to the master.

The report data on the poller will overwrite the data on the master system

More information

For more information and a more complex example please take a look at the howto in the git repository of the opensource project of Merlin:

http://git.op5.org/git/?p=nagios/merlin.git;a=blob;f=HOWTO;hb=master#l171

op5 Monitor API and CLI

About op5 Monitor API

This chapter covers the following topics:

Subject	Page	Subsections
Introduction	170	
GUI API	171	
Configure API	172	*To exectue the op5 Monitor CLI* on page 173

Introduction

op5 Monitor comes with a few APIs that can be used to The following APIs can be used.

- Ninja API, GUI API
- Nacoma API, Configure API
- op5 Monitor Configuration CLI

GUI API

The GUI API is use to get the information that is used by op5 Monitor GUI. It can give you information about all objects used by the op5 monitor.

Widgets are one place where the GUI API will come handy.

There are only a breifly documentation about that API today. It is included in the product.

Let us say that your monitor server is called op5-monitor you can reach the documentation on the following location:

https://op5-monitor/monitor/Documentation/html/index.html

It is generated by doxygen and contains information like

- namespaces
- structures (classes and methods)
- files

Configure API

The configure API is used to manipulate the object configuration used by op5 Monitor. It works against the configure database the same way as the op5 Monitor Configuration tool does.

You may use it to build integrations between op5 Monitor and other third party software.

Let us say that your monitor server is called op5-monitor you can reach the documentation on the following location:

https://op5-monitor/monitor/op5/nacoma/Documentation/html/index.html

It is generated by doxygen and contains information like classes and methods used in the op5 Monitor configuration tool.

op5 Monitor Configuration CLI

This is a tool used to edit the op5 Monitor object configuration.

You may use this one to add and remove

- hosts
- services
- contacts
- timeperiods.

You may also

- list objects
- save configuration
- undo configuration (force import of the config files to the configure database).

Exectuing the op5 Monitor CLI

To exectue the op5 Monitor CLI

1. Logon to the op5 Monitor server as root
2. Exectute the following command:
   ```
   php /opt/monitor/op5/nacoma/api/monitor.php
   ```

The above example will give you a description about how to use the op5 Monitor Configuration CLI

Wiki

About Wiki

This chapter covers the following topics:

Introduction

In op5 Monitor there is a wiki included witch can be used for documenting hosts and services.

Wiki pages can be created for both hosts and services.

The wiki has a built in version revisioning, this can be used to track changes in wiki pages and restore an older version. It is also possible to view changes between versions.

The wiki can be used for documenting hardware information, serial numbers and other information regarding a host or service. It can also be used to document workflows and how to act when there is a problem with a host or service.

The official docuwiki manual can be found here: http://www.dokuwiki.org/manual

Managing wiki pages

Create a wiki page

To create a wiki page for a host or service

1. Go to ‘Configuration’
2. Go to the host or service you want to create a page for.
3. Click on ‘Advanced’
4. Scroll down to ‘notes_url’ and click ‘Use wiki’. This will add a notes url to a wiki page. [SCREENSHOT]
5. Click ‘Apply Changes’ and save your configuration.
6. Go to the host in op5 Monitor and click on ‘Extra notes’ [SCREENSHOT]
7. Click on ‘Create page’.
8. Edit the information and click on ‘Save’

Deleting a wiki page

If you edit a page and remove all its content then DokuWiki will delete the page, and the associated page name.

For more information about the docuwiki

http://www.dokuwiki.org/manual

Third party configuration import

About Third party configuration import

This chapter covers the following topics:

Introduction

Op5 Monitor has the capability to import the configuration from an nagios installation.

To do follow this manual basic knowledge in linux and nagios is necessary.

Pre-requirements

A running nagios 3.x installation and op5 Monitor.

Limitations

There are some of limitations of the import script.

- The import-script does not work with a nagios 1 or 2 installation.
- Host and service history can not be imported, but can be copied manually.
- Graph history can not be imported.

Import configuration

To import a nagios 3 configuration we need to prepare the nagios configuration files first, after that we can use the import script to import the files into op5 Monitor.

Preparing nagios configuration

Log in to the nagios server via ssh or locally.

Create a new file called templates.cfg in which you manually add both your host-templates and your service-templates. These are usually located in hosts.cfg and services.cfg.

Create a nagios pre-cache file by stopping nagios and start it with the -p option. this is done from you nagios binary directory, usually “/usr/local/nagios/bin/”.

```
# service nagios stop
# ./nagios -pv <path to your nagios.cfg>
```

This will create a file called objects.precache in your “var” directory under your nagios installation.

Import nagios configuration

Make sure op5 monitor is stopped

```
# mon stop
```

Copy the files to the correct directory on your op5 Monitor server.

File	To folder
objects.precache	/opt/monitor
templates.cfg	/opt/monitor
nagios.log	/opt/monitor/var/
log archive	/opt/monitor/var/archives

Run the import script

```
# php /opt/monitor/op5/nacoma/import-reduce.php --cfg-file=/opt/
monitor/templates.cfg --object-cache=/opt/monitor/
objects.precache
```

Do a config-test on the imported configuration

service monitor configtest

If you have any errors these needs to be resolved before we can continue with starting the op5 monitor service.

When there are no issues left start the monitor service

```
# mon start
```